Fachhochschule
Dortmund
University of Applied Sciences and Arts

Setup of test environments based on a Xilinx Zynq SoC for measuring the leakage current and for radiation qualification of SRAM based FPGAs

Master Thesis

Fachhochschule Dortmund
Energy Systems

Lars Koers

April 2024

Thesis committee: Prof. Dr.-Ing. Michael Karagounis
Felix Schneider

Declaration

Hiermit versichere ich, dass die von mir vorgelegte Prüfungsleistung selbständig und ohne unzulässige fremde Hilfe erstellt worden ist. Alle verwendeten Quellen sind in der Arbeit so aufgeführt, dass Art und Umfang der Verwendung nachvollziehbar sind.

Dortmund, April 25, 2024

Unterschrift

Abstract

This thesis discusses the development of test environments using Xilinx Zynq System on Chip (SoC) for measuring leakage currents and radiation qualification of Static Random Access Memory (SRAM) based Field Programmable Gate Arrays (FPGAs) at European Organisation for Nuclear Research (CERN). The effects of radiation on electronic components are explained, followed by an introduction to the FPGAs used. The GateMate FPGAs leakage current is measured in its application area with respect to temperature and core voltages. A comparable testing environment is used from the tester to the tested device, as it will later be used at CERN. The GateMate is being prepared in this setup for the finalization of radiation qualification at CERN, to be transferred later. For this purpose, the basic tests are explained and the outstanding tests are then carried out. The Lattice iCE40 UltraLite FPGA is used in an initial application test to determine its suitability for further radiation qualification tests at CERN. The analysis and presentation of the test results are followed by a summary and outlook.

Diese Arbeit behandelt die Entwicklung von Testumgebungen mit einem Xlinix Zynq SoC zur Messung von Leckströmen sowie der Strahlenqualifizierung von SRAM basierten FPGAs am CERN. Zunächst werden die Effekte von Strahlung sowie deren Wirkung auf elektronische Komponenten erläutert. Anschließend wird eine Einführung in die verwendeten FPGAs gegeben. In der Durchführung wird der Leckstrom des GateMate FPGAs in seinem Einsatzbereich hinsichtlich Temperatur und Core-Spannungen gemessen. Es wird eine Testumgebung mit ähnlichen Verbindungen zwischen Tester und Testgerät verwendet, wie sie später bei Bestrahlungsstudien am CERN Anwendung findet. Der GateMate wird in diesem Setup für die Finalisierung der Strahlenqualifikation vorbereitet, um die spätere Inbetriebnahme am CERN zu beschleunigen. Dazu werden die Tests grundlegend erläutert und die noch ausstehenden im Anschluss durchgeführt. Der Lattice iCE40 UltraLite FPGA wird in einem ersten Applikationstest eingesetzt, um seine Eignung für weitere Tests zur Strahlenqualifikation am CERN zu bestimmen. Die Ergebnisse aller durchgeführten Tests werden ausgewertet und dargestellt. Abschließend wird eine Zusammenfassung mit Ausblick präsentiert.

Contents

Acronyms

ALICE A Large Ion Collider Experiment

ASIC Application Specific Integrated Circuit

ATLAS A Toroidal LHC ApparatuS

BRAM Block RAM

CERN European Organisation for Nuclear Research

CHARM CERN Highly Accelerated Mixed Fiel Facility

CLB Configurable Logic Block

CMS Compact Muon Solenoid

COTS Commercial Of The Shelf

CPE Cologne Programmable Element

CRAM Configuration RAM

DD Displacement Damage

DFF D-style Flip-Flop

DMM Digital Multimeter

DPSRAM Dual Port SRAM

DUT Device Under Test

EBR Embedded Block RAM

ECC Error Correction Code

FHDO Fachhochschule Dortmund

FIFO First In First Out

FMC FPGA Mezzanine Card

FPGA Field Programmable Gate Array

FTDI Future Technology Devices International

GHDL G Hardware Design Language

GPIO General Purpose Input Output

HDL Hardware Description Language

IEL Ionizing Energy Loss

JTAG Joint Test Action Group

LC Logic Cell

LHC Large Hadron Collider

LHCb Large Hadron Collider beauty

LED Light Emitting Diode

LET Linear Energy Transfer

LUT Look Up Table

LXI LAN eXtension for Instrumentation

MBU Multiple Bit Upset

MEU Multiple Event Upset

MOSFET Metal-Oxide Semiconductor Field-Effect Transistor

NIEL Non Ionizing Energy Loss

NVCM Non Volatile Configuration Memory

PCB Printed Circuit Board

PIF Proton Irradiation Facility

PLB Programmable Logic Block

PLL Phase Locked Loop

PMOD Peripheral Module

POR Power-On Reset

PSI Paul Scherrer Institute

PUF Physically Unclonable Function

RAM Random Access Memory

RHA Radiation Hardness Assurance

SEB Single Event Burnout

SEE Single Event Effect

SEFI Single Evenet Functional Interrupt

SEGR Single Event Gate Rrupture

SEL Single Event Latch-up

SET Single Event Transient

SEU Single Event Upset

SoC System on Chip

SPI Serial Peripheral Interface

SRAM Static Random Access Memory

STI Shallow Trench Isolation

TID Total Ionizing Dose

TMR Triple Modular Redundancy

USB Universal Serial Bus

VHDL Very High Speed Integrated Circuits (VHSIC) Hardware Description Language

WSR Window Shift Register

Nomenclatur

Variable	**Definition**	**Unit**
C	Capacitance	F
C_{ox}	Oxide Capacitance	F
E	Energy	eV
f	Particle flux	p/cm^2s
I_D	Drain current	I
LET	Linear Energy Transfer	MeV/cm
p	Density	kg/m^2
q	Charge	C
S	Stopping power	MeV/gcm^2
t	Time	s
V_{TH}	Threshold voltage	V
V_{GS}	Gate-Source-Voltage	V
V_ϑ	Thermal voltage	V
x	Distance	cm
Φ	Particle fluence	p/cm^2

1 Introduction

How does the world work and what keeps it all together? Scientists around the world are asking themselves questions like these, and some of them are being explored at the European Organization for Nuclear Research (CERN), to find answers. CERN is the largest particle physics laboratory in the world. In different facilities particles are accelerated to let them collide with each other or on a target. An overview of the whole accelerator complex in 2022 of CERN can be seen in figure 1.1.

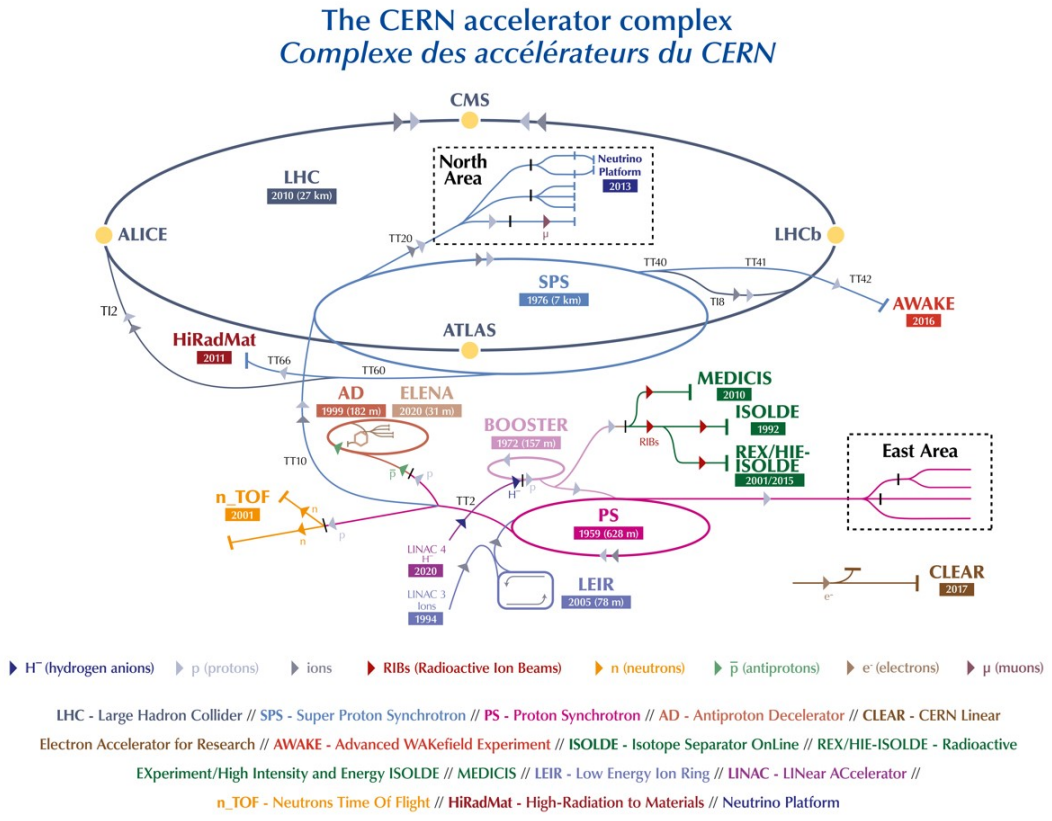


Figure 1.1: CERN Accelerator Complex[24]

The most famous and largest one is the Large Hadron Collider (LHC) [6]. High energy

particle beams are accelerated to let them collide in one of four experiments. These are A Toroidal LHC ApparatuS (ATLAS), A Large Ion Collider Experiment (ALICE), Compact Muon Solenoid (CMS) and Large Hadron Collider beauty (LHCb). ATLAS and CMS are used for studying the standard model and for example particles that make up dark matter. These two experiments led to the discovery of the Higgs boson in 2012 [1]. ALICE is used to study the physics of strongly interacting matter at extreme energy densities, which forms quark-gluon plasma [5]. At the LHCb the differences between matter and antimatter are studied.
To ensure error-free operation of all the systems, the electronics which are used for control and measuring have to be able to work without failure under harsh conditions. Depending on where parts are placed in the accelerator complex, they are disposed to various levels of radiation. This can effect the electronics in different ways. Therefore it is necessary to characterize a devices behaviour and tolerance under irradiation. It would be costly and time consuming to produce specially designed devices to withstand exposure in all of these areas. To reduce costs, Commercial Of The Shelf (COTS) electronics are used. CERN is testing parts in different facilities to find out if they will work under the expected conditions and when they need to be replaced to prevent failure.
Field Programmable Gate Array (FPGA)s are devices, that are used extensively in accelerator electronics. Therefore these are being tested to evaluate their potential for use in different accelerator areas. FPGA's are COTS that are widely spread on the market and very flexible in use. These devices can be freely configured for the required use case. By hardware design in a Hardware Description Language (HDL) the system can be optimized for its task and even reused for a new task by rewriting the configuration with a new design. This allows the usage of one device type for many different tasks, while an Application Specific Integrated Circuit (ASIC) for example can only be used for its one use case.
For the qualification of parts and devices at CERN, the Radiation Hardness Assurance (RHA) process is performed [7]. This process includes seven phases which are shown in figure 1.2.

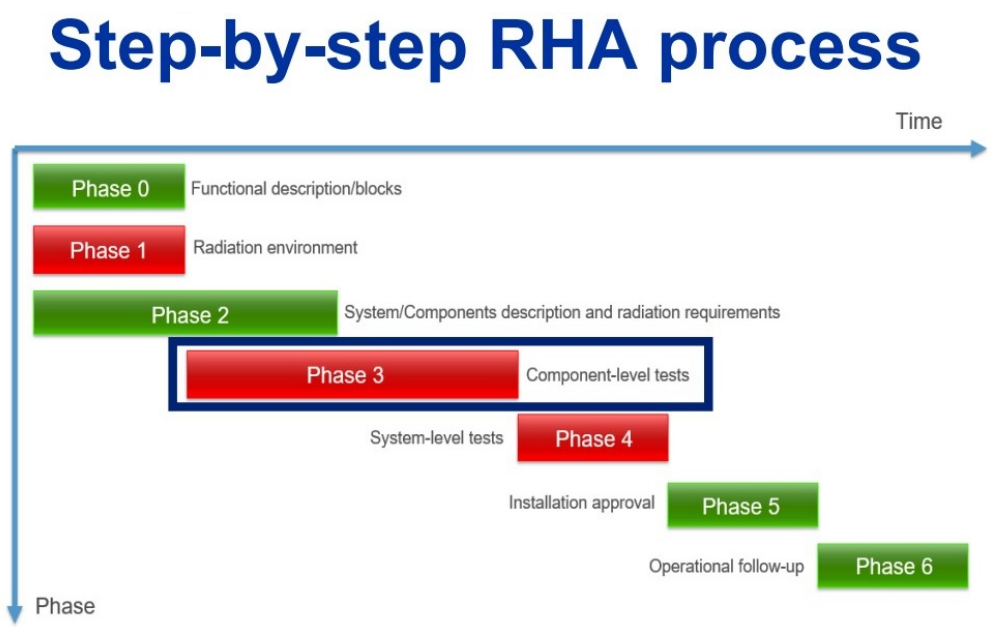


Figure 1.2: Phases of the RHA process at CERN[7]

At the beginning of the RHA process, during phase zero, the specifications must be established. The expected radiation exposure is determined in phase one. In phase two, the maximum failure rate for both the system and its components per year and lifetime are defined. Subsequently, different components are subjected to initial setups for SEEs or Displacement Damage (DD) testing. These assessments are carried out at both internal and external facilities. In phase four a system is tested in a mixed radiation field available at CERN's CERN Highly Accelerated Mixed Fiel Facility (CHARM), which provides a comparable environment to the expected conditions in the LHC area. Only after the successful completion of these previous steps will the system be deployed in phase five and monitored for performance in phase six. For this master thesis, phase three is the most relevant. The FPGAs to be tested will be prepared for this step of the RHA and the results evaluated.

The first goal of this thesis is to become familiar with the GateMate [8] FPGA and to replicate the irradiation test setups from previous campaigns at CERN as documented in [14]. The focus will be on achieving technical proficiency and reproducibility. Two outstanding tests, that were not finished previously, must be performed. The already collected data of the TID-test will be evaluated with focus on the relation of frequency between the implemented Ring Oscillators over radiation dose. Furthermore the Lattice iCE40 UltraLite [30] FPGA will be introduced and an application test will be carried out to see if this type of FPGA could be of interest for future projects at CERN.

An overview of general radiation effects is given first. This is followed by a brief explanation of the expected effects of radiation on FPGAs. The test preparation setup includes two FPGAs. The GateMate integrated on an evaluation board [11] and a Xilinx FPGA integrated on a Zedboard [28]. The GateMate will be the FPGA under test, while the Zedboard will serve as the controlling device. For testing at CERN the controlling device is the Microzedboard [17] and the tested devices are the GateMate and the iCE40 UltraLite integrated on the Breakout Board. In addition, the required software and hardware is documented with basic information.

For the project's initiation and to familiarize oneself with the GateMate evaluation board, as well as to acquire basic proficiency in available software libraries a test is conducted to measure the GateMate's leakage current. The GateMate is tested at various temperatures and core voltages while it is kept in a reset state. The test setup consists of a Zedboard, power supply, multimeter, and Thermostream. For radiation qualification at CERN, a basic explanation of the tests previously performed with the GateMate is provided, outlining their set up and functionality. Two outstanding tests are performed and their setups are explained in more detail with the evaluation of the results. Finally, the Lattice iCE40 FPGA is introduced in an application test, which is explained as well and the results are evaluated in terms of further research.

2 Radiation effects and their impact on electronics

When electronics are exposed to radiation, various effects can occur and influence their behaviour. The following chapter describes the basics of these effects. First, the general effects of radiation on matter are discussed, followed by the effects that occur in electronic components. These general effects are also described in [22] and the effects on related electronics are described in [14] and [23].
Relevant physical quantities in this context are flux, fluence, and dosis. The dosis represents the energy deposited by ionizing radiation and corresponds to the absorbed energy per kilogram of matter, denoted as $Gy[J/Kg]$. Flux and fluence are related, with flux referring to the amount of particles passing through an area per unit time, denoted as $f[p/cm^2s]$. The fluence representing the integral of the flux over a time period, denoted as $\Phi[p/cm^2]$.

2.1 Interaction with matter

Depending on the material and the type of particles, different effects can occur. These are then also dependent on the energy of the impacting particle. Such effects may be classified as those induced by photons, neutrons, or charged particles. Figure 2.1 illustrates the potential interactions arising from these particles.

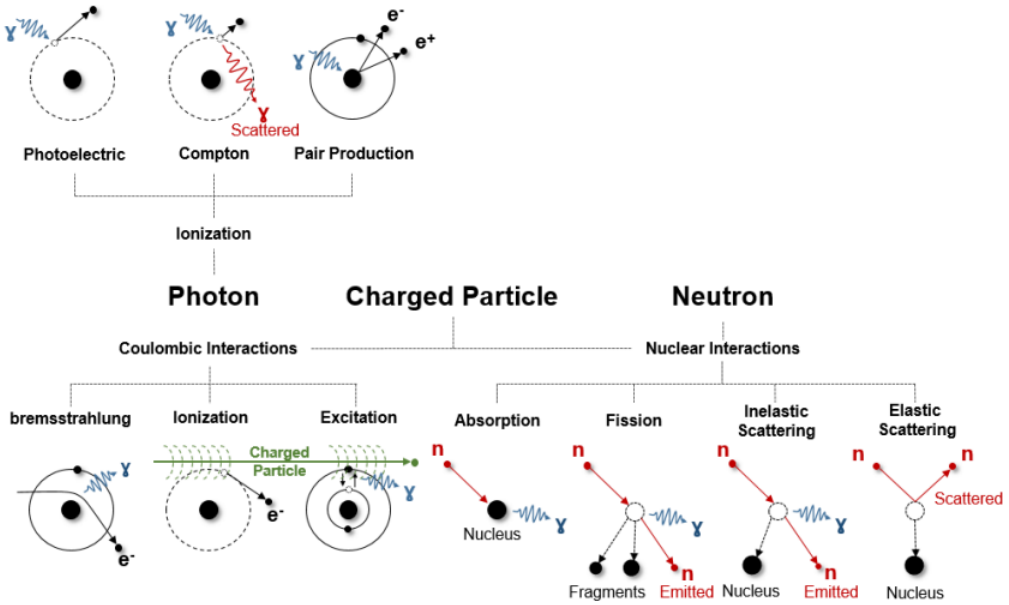


Figure 2.1: Radiative interactions[22]

The interactions with photons are the following:

- **Photoelectric effect**
  The photons energy has to be close or similar to the binding energy of an orbital electron. When they interact, all the energy of the photon is given to the electron, which in effect is ejected from the atom.

- **Compton effect**
  The photons energy has to be much greater then the binding energy of the electron. In this case, some of the photons energy is transferred to the electron, which in effect is ejected from the atoms orbit. The photon looses some energy and gets scattered.

- **Pair production**
  The photons energy has to be higher than 1MeV and it has to pass very close to the nucleus of the atom. Then the photon may lose all of its energy, which effects in an electron/positron pair.

Neutrons can only have nuclear interactions, which are displayed at the right of 2.1.

- **Neutron absorption**
  The neutron (mainly thermal neutrons with an energy of about 25meV) gets absorbed by the nucleus, which then is in an excited state. By emitting one ore more photons, the nucleus leaves this state.

- **Fission**
  The neutron gets absorbed by a heavy nucleus, which then splits into smaller nuclei. In this effect, energy is released in form of neutrons and photons.
- **Elastic scattering**
  The neutron collides with a nucleus and gets scattered. Some energy is transferred to the nucleus, which also scatters.
- **Inelastic scattering**
  The neutron gets absorbed by a nucleus. In effect it emits a neutron with lower energy and is in an excited state. Similar to neutron absorption, the neutron leaves this state by emitting one ore more photons.

With heavy charged particles, both nuclear interactions and coulombic interactions can take place. Examples for these are particles with a charge unequal to zero, like pions ($\pi^{\pm}$) , protons ($p^{\pm}$), kaons ($K^{\pm}$) or muons($\mu^{\pm}$).

- **Bremsstrahlung**
  When a light charged particle passes close to a nucleus, it becomes deflected. Some of its energy is converted to a photon.
- **Ionization**
  When a charged particle with high enough energy passes close to an orbital electron, the electron is ejected from its orbit
- **Excitation**
  When a charged particle passes close to an orbital electron, the electron can move between the inner shells and release photons in this process. The energy of the particle has to be lower than the binding energy of the electron.

The ionization of an atom is called *direct ionization*, if the interaction happens directly from the striking particle. Newly generated particles can than also ionize atoms, which then is called *indirect ionization*. In effect of these interactions, energy loss is taking place until the particle is completely stopped. The energy loss can be categorized into two groups.

- **Ionizing Energy Loss (IEL)**
  IEL is the energy which is lost in effect of the interactions causing ionization.
- **Non Ionizing Energy Loss (NIEL)**
  NIEL is the energy which is lost in the nuclear interactions, causing DD.

The Linear Energy Transfer (LET) is the rate of energy loss along the path of the particle. It can be described by the following equation 2.1.

$$LET = \frac{dE}{dx} \tag{2.1}$$

By normalizing this equation to the material density ρ, the mass stopping power can be described with 2.2.

$$S = \frac{1}{\rho}\frac{dE}{dx} \tag{2.2}$$

When passing through matter, the LET value is strongly dependent on the material and the particle. Most of the energy is released just before the end of its way. This point is also called Bragg Peak. The total energy that is emitted and absorbed by the material is called Total Ionizing Dose (TID) and can be determined with the formula 2.3 and use of the unit conversion parameter K, with the value $K = 1.6x10^{-}7$ Gy g MeV.

$$TID = K \int \frac{LET}{\rho}\Phi(\rho, E)dE \tag{2.3}$$

2.2 Impact on electronics

All possible radiation effects have the potential to impact electronics. The effects range from brief, unperceived events to damage that renders the device inoperable. The effects categorize into the following types.

- **Cumulative effects**
  Every effect on the device lead to a degradation and is accumulated up over the whole lifetime. So for example the current consumption is rising more and more over a long time until the device just stops working.

- **Single Event Effect (SEE)**
  This effect has a singular impact on the behaviour of the device and can be classified as either destructive and non-destructive.

In order to keep the focus on the relevant topic, the following effects are related to CMOS technology, as the Device Under Test (DUT) was manufactured using this technology.

2.2.1 TID Effects

TID Effects occur due to the ionization of insulation materials, specifically the silicon dioxide film used in Metal-Oxide Semiconductor Field-Effect Transistors MOSFET. This ionization generates electron-hole pairs that result in the accumulation of charges, leading to degradation of device performance. It should be noted that this effect is more pronounced in Metal-Oxide Semiconductor Field-Effect Transistor (MOSFET)s with thicker gate oxides, i.e. those with thickness greater than 5 nanometers, where the affected layer lies between the gate and the SiO_2/Si interface. Modern technologies are less susceptible to this effect because they employ smaller gate oxides. For these types of devices, TID effects occur within the Shallow Trench Isolation (STI) oxide, which isolates the transistors from each other. The accumulation of charges can cause current to flow between

the drain and source, resulting in an increase of leakage current, which is illustrated in Figure 2.2. This effect impacts the overall power consumption of the device.

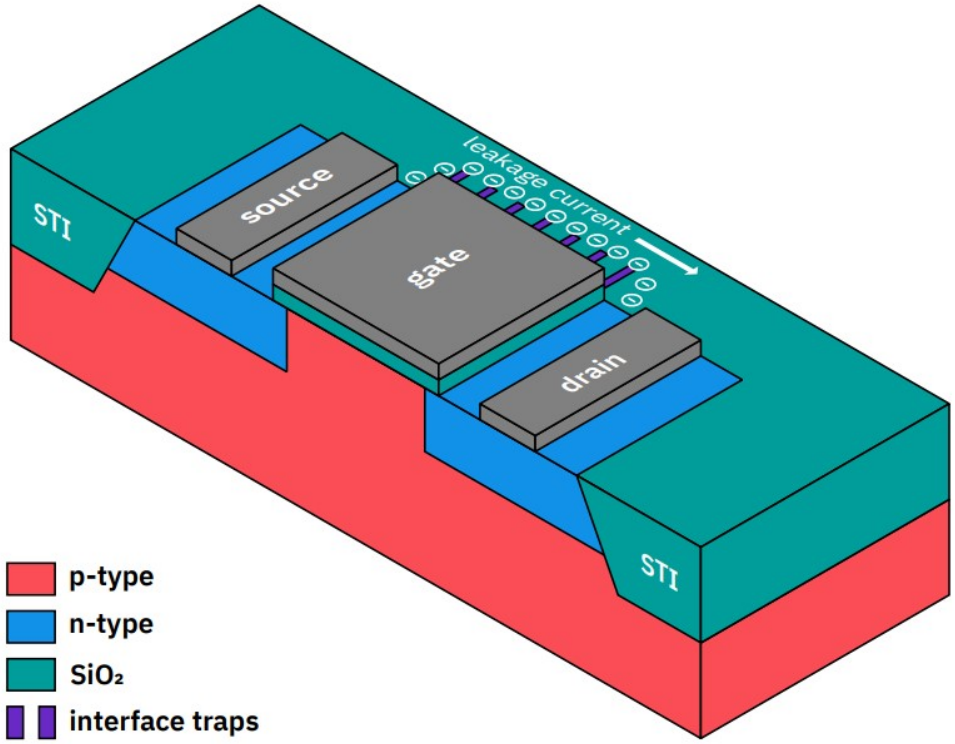


Figure 2.2: TID effects leading to a leakage current in the STI oxide [14]

MOSFETs with thicker gate oxides, see figure 2.3, are still used for input and output interfaces in FPGAs, due to their better tolerance to higher voltages. Here the TID can result in a shift of the threshold voltage. The following example for this process take place in the oxide of an n-channel MOSFET.

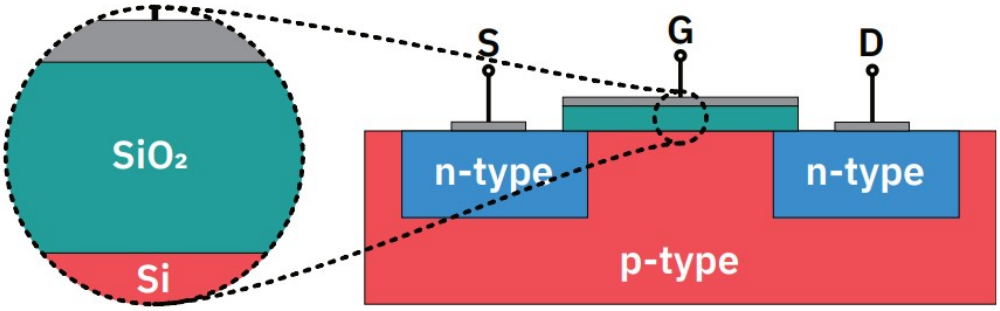


Figure 2.3: N-Channel MOSFET structure [14]

- 1. Electron-hole pairs are generated by the striking particle (amount dependent from its energy). The electrons will flow in direction of the gate, while the holes flow towards the SiO_2/Si interface. Some electrons may recombine on their way with holes. The fraction of electron-hole pairs that not recombine and escape is called the charge yield.
- 2. The electrons escape trough the gate, while the holes drift trough the SiO_2 layer in direction of the substrate.
- 3. Holes travelling trough the oxide can be trapped and lead to oxide traps. Furthermore holes release protons while travelling, which can be trapped at the SiO_2/Si interface and so are called interface traps.
- 4. Both traps are resulting in a shift of the threshold voltage. The interface traps additionally reduce the mobility of the charge carriers.

Figure 2.4 shows the explained effects from left to right comparing to point one to three.

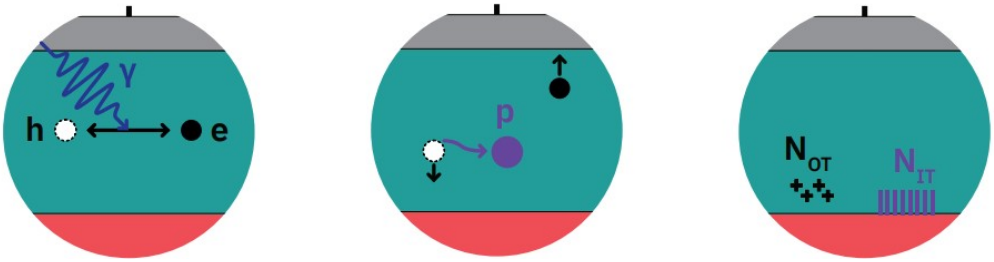


Figure 2.4: TID effects in the gate oxide [14]

The build-up of charge has a different impact on pMOS and nMOS transistors:

- **nMOS**
  It will be easier to create an inversion channel with the already build charge.
  Turn on voltage is lowering -> until threshold voltage becomes negative and it is not possible to turn off.

- **pMOS**
  A higher voltage is needed at the gate to create an inversion channel.
  Turn on voltage needs to be higher -> until threshold voltage is higher than the applied supply voltage and it is not possible to turn on.

The shift of the threshold voltage can be described by the following equation:

$$\Delta V_{TH} = -q\frac{1}{C_{ox}}\Delta N = V_{OT} \pm -q\frac{1}{C_{ox}} = V_{IT} \tag{2.4}$$

The C_{ox} is the oxide capacitance, ΔV_{OT} stands for the oxide-trapped and the ΔV_{IT} for the interface-trapped charge. The plus or minus in the second term depends on the type of the MOSFET. For p-channel the minus and for n-channel the plus is used in the equation 2.4.

2.2.2 Single Event Effect (SEE)

SEE effects are divided between destructive SEEs, which possibly permanently damage a device and non-destructive SEEs, which can potentially be undone by resetting the device. Destructive SEEs are:

- **Single Event Latch-up (SEL)**
  If a low impedance path between power supply and ground of the device is created, a higher current can occur and cause a fatal damage. The effect can be stopped by a power cycle.

- **Single Event Burnout (SEB)**
  When a heavy-ion cause a second breakdown in a MOSFET, the device can fail because of thermal runaway and the resulting high current.

- **Single Event Gate Rrupture (SEGR)**
  A heavy-ion can cause a conducting path in the gate oxide. This results in a larger field across the dielectric, until it exceeds a limit and causes gate rupture.

Non-destructive SEEs are:

- **Single Event Upset (SEU)**
  The particle that hits the device, has an energy high enough to invert a logic state of a storage element. By rewriting the memory this error can be corrected.
- **Multiple Event Upset (MEU)**
  In this case, multiple storage elements close to each other can be affected at once, because of the lateral diffusion. This effect is also called Multiple Bit Upset (MBU).
- **Single Event Transient (SET)**
  A voltage spike can propagate through the device for a short time. It is possible that it is not even noticed.
- **Single Evenet Functional Interrupt (SEFI)**
  An ion particle can cause a temporary failing state. It is possible that the state recover after a while, or the systems needs a reset.

Further explanations, as well as test procedures for SEEs can be found in [4].

2.3 Mitigation techniques for radiation effects

Whenever an SEE occurred, it is possible to undo it by rewriting the corrupted storage element. This is only feasible if the effects are non-destructive. However, it is still possible for a single effect to cause the entire system to fail depending on the running setup. As a result, engineers have developed various methods to minimize these impacts [16].

Memory scrubbing
Like already mentioned, it is possible to rewrite an effected memory in case of an upcoming event. To check if the configuration is corrupted, one approach is the so called memory scrubbing. The configuration is stored in a second independent memory. By an external configuration manager, the memory of the device is read continuously to compare it with the reference in the secondary memory. In the case of a difference, the configuration will be rewritten. This process avoids the accumulation of errors that could result in design failure.

Error Correction Code (ECC)
This approach checks a memory by the use of channel codes like the Hamming code to detect bit flips. Furthermore a detected flips are corrected. The GateMate FPGA provides this feature for checking its memory blocks to detect single flipped bits and to correct them as well. Two bit errors can only be detected. This technique is explained

in [14], but was not implemented for the tests.

Triple Modular Redundancy (TMR)
One of the most common techniques, which is used for the test setups in this thesis as well, is triple module redundancy (TMR). For this approach, each flip-flop or even a whole module is set up three times. The level of triplication can for example be decided by the availability of resources. Depending on the implementation technology, the TMR can be integrated by software automatically or manually by the user, which is then referred to as block TMR. However, manual integration is often less effective then integration by software. The same input is used three times, and the output is determined by a voter. This voter chooses the correct output by majority, if one of the three values is affected by an SEU. The structure of this setup is shown in figure 2.5.

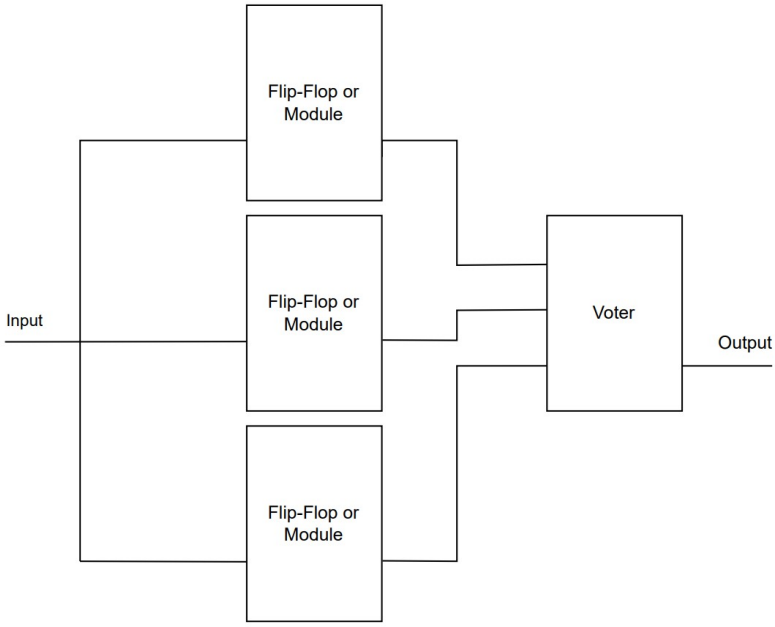


Figure 2.5: Principle of the Triple Modular Redundancy (TMR)

These techniques can optimize a design for the expected conditions. Other techniques are also available, but for this test TMR is chosen and implemented at block level for comparison of the characteristics of an unprotected design. If the system fails during testing, the design will be rewritten and the device will be restarted. As previously stated in the introduction, a device must undergo the RHA process at CERN. Step three consists of preparing component-level tests for the device, which will be described in chapter 7.

3 Leakage current and temperature

As mentioned in chapter 2.2.1, the leakage current of a MOSFET increases with TID and can cause device damage. Temperature also affects the leakage current [18]. As the temperature rises, the threshold voltage of a MOSFET decreases and the charge carrier mobility is reduced. This can result in two opposing effects. When the gate-source voltage is low, the lowering of the threshold voltage has an higher impact than the lowering of the mobility, resulting in an increase in current. However, at high gate-source voltages, the lowering of mobility dominates and the current decreases. It is important to note that the leakage current is measured in a reset state, with the objective of differentiating between dynamic and leakage current consumption. In the reset state, only the N-MOS or the P-MOS transistors of a CMOS structures are turned on. As temperature rises, the transistor turned off becomes more conductive, resulting in an increase in leakage current.

The equation 3.1 describes the threshold voltage (V_{TH}) with T representing the temperature and the zero suffix indicating the operating point of a value. Alpha is the temperature coefficient, which is negative for a silicon structure.

$$V_{TH}(T) = V_{TH0} * (1 + \alpha * (\Delta T)) \tag{3.1}$$

When the temperature increases, ΔT increases as well. Since alpha is negative for silicon, the threshold voltage decreases as the temperature rises. Additionally, the threshold voltage affects the drain current (I_D), as described in equation 3.2, where V_{GS} is the gate-source voltage and V_ϑ is the thermal voltage. The zero suffix refers to a value at the operating point.

$$I_D = I_{D0} * e^{\frac{V_{GS}-V_{TH}}{V_\vartheta}} \tag{3.2}$$

It is evident that as the threshold voltage decreases, the value in the exponent of the e-term increases, causing the e-term to become larger. As a result, the drain current increases, which should ideally be zero for low power consumption. This effect significantly impacts the static power dissipation of a device. To demonstrate this behavior the leakage current of the GateMate FPGA is measured across its entire specified temperature range.

4 FPGA basics

FPGAs are integrated circuits that can be configured for a wide range of tasks. By the use of a Hardware Description Language (HDL) a hardware design can be generated to describe the needed functionality which should be implemented. FPGAs consist among other things of input and output interfaces, clocking sources and configurable logic blocks CLBs, which are interconnected in a routing structure. One of the significant advantages is the possibility of reconfiguration, allowing the optimization of a single device for multiple tasks.

At CERN FPGA-devices are widely used. They are COTS that are flexible in their application, have high performance and are less cost intensive in acquisition and replacement than special build products. For the accelerators many FPGAs are used to analyze process data. Due to the variable radiation levels in the accelerator environment, an understanding of the behaviour of FPGAs under such conditions is critical for safe operation and selection of the right devices or countermeasures.
Several companies manufacture various FPGAs with different technologies. Three common types are antifuse, flash memory and static memory FPGAs [3]. Antifuse types can be described as a lot of "open switches" in the initial state. With a high current these can be "burned" to a closed state. In this way a configuration can be written in the device. While this process is permanent, the other two types are rewritable.
The flash memory can keep its configuration, but needs a high voltage to be configured. The structure in the device is similar to a MOSFET, but has a different gate setup. A second floating gate is placed below the normal gate. Because of the fact, that the floating Gate is isolated, electrons that are placed on it by means of a strong electric field can not escape. In this way the state is kept until a new electric field configuration is applied.
An SRAM cell consists out of two inverters in a feedback loop and two pass transistors to write a binary value. This structure does not require a high voltage or current to be written, but is loosing its configuration, every time the power is turned of and so has to be reconfigured.
The subsequent SRAM type FPGAs are intended for radiation qualification in different tests.

4.1 GateMate FPGA

The GateMate [9] is a D-latch based FPGA that is designed by the company Cologne Chip AG [1] in Germany. The chip is manufactured in a 28nm CMOS technology by GlobalFoundries [2] in Germany as well. This FPGA is part of the radiation tests and has also been the subject of tests in various setups in the year 2022. The methods and results obtained so far, together with a detailed device description, can be found in [14].

For the tests, the FPGA CCGM1A1 is utilized on the evaluation board from Cologne Chip [11]. Figure 4.1 shows an overview of the board's features.

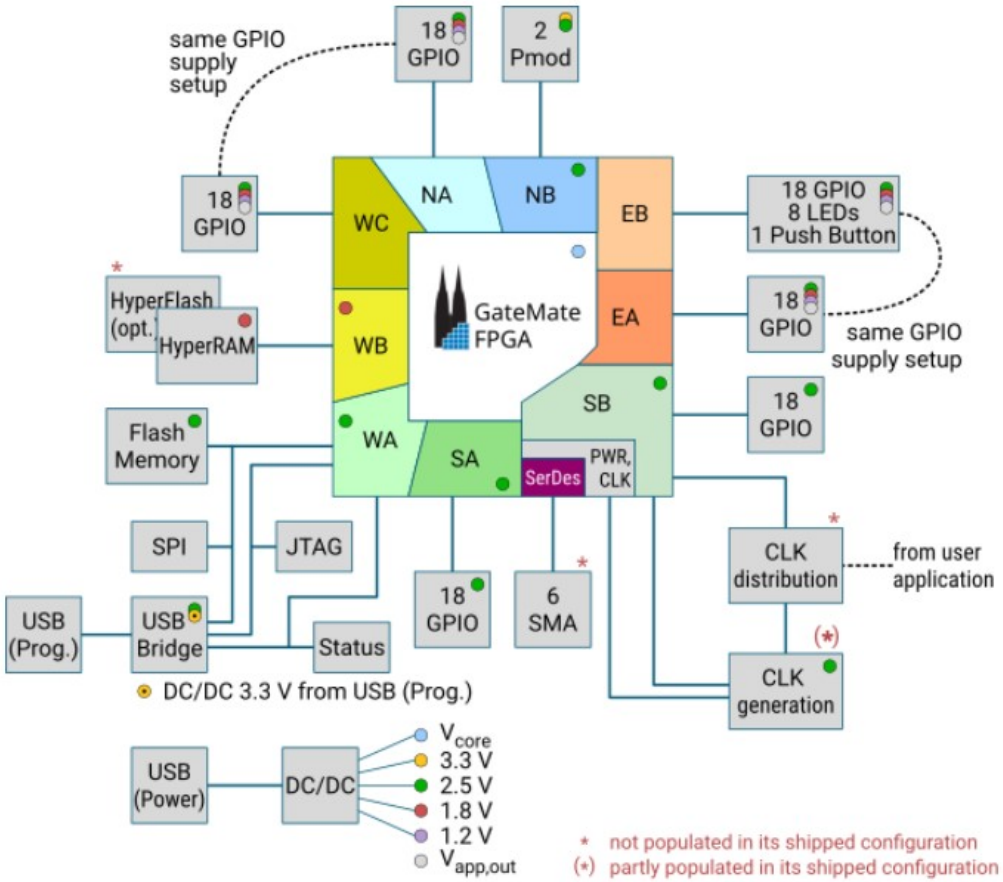


Figure 4.1: Overview of the evaluation board features [8]

[1] https://colognechip.com/
[2] https://globalfoundries.com/

Features that are mainly used in the tests are as follows,

- 20,480 Configurable Logic Block (CLB)s, in this device referred as Cologne Programmable Element (CPE)s,
- 0,9-1,1V core voltage for different core power,
- 4 clock generators(PLL),
- 32 SRAM blocks with each 40KiB,
- 6 General Purpose Input Output (GPIO) banks with 1,2-2,5V,
- 2 Peripheral Module (PMOD) connectors with 2,5-3,3V.

The figure 4.2 shows a top view of the used evaluation boards. The FPGA itself is close to the middle of the device, with the logo of the company on it. It is surrounded by the 6 GPIO-banks. On the left side, there are the connectors for the power supply while in the upper area there are voltage regulators situated next to the two PMOD-connectors.

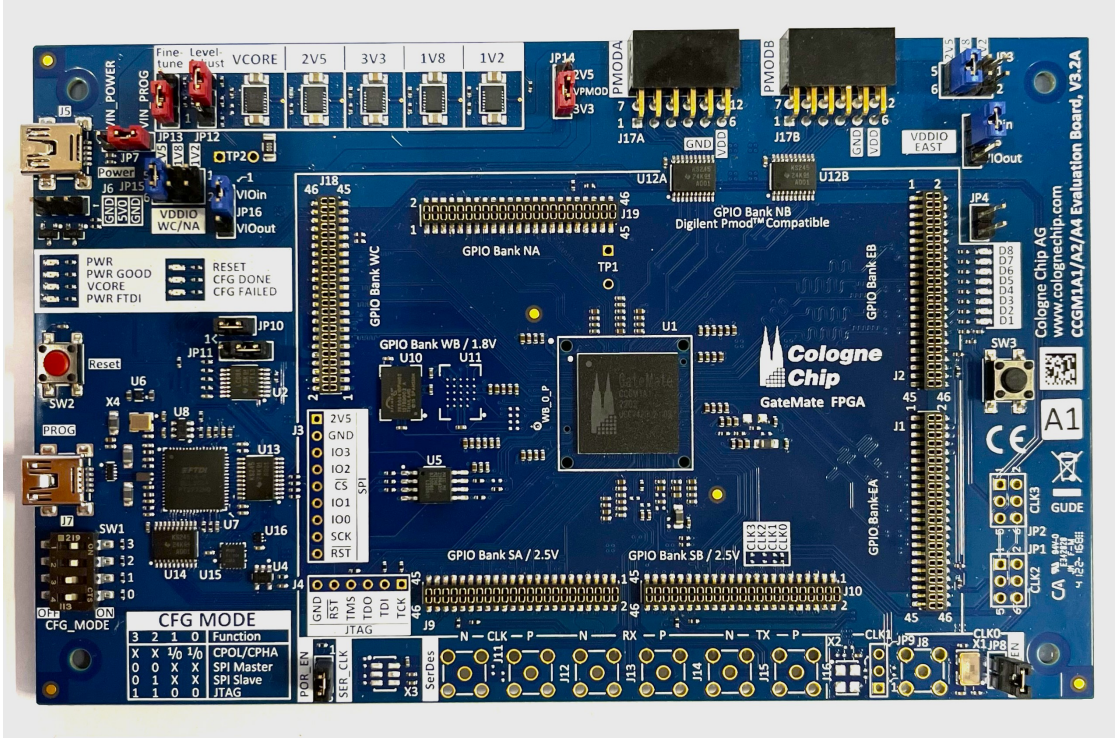


Figure 4.2: Top view of the GateMate evaluation board

The FPGA is build up out of so called Cologne Processing Elements CPEs. Each containing a Look Up Table (LUT)-tree with eight inputs, two flip-flops, fast signal routing paths and two outputs. Figure 4.3 displays the structure of a CPE. The LUT may be used as a single 8-input LUT or two 4-input LUTs. Additionally the CPE can be set up as 1-bit or 2-bit full adder, 2x2-bit multiplier or as 6 inputs with 4-input multiplexer function [8].

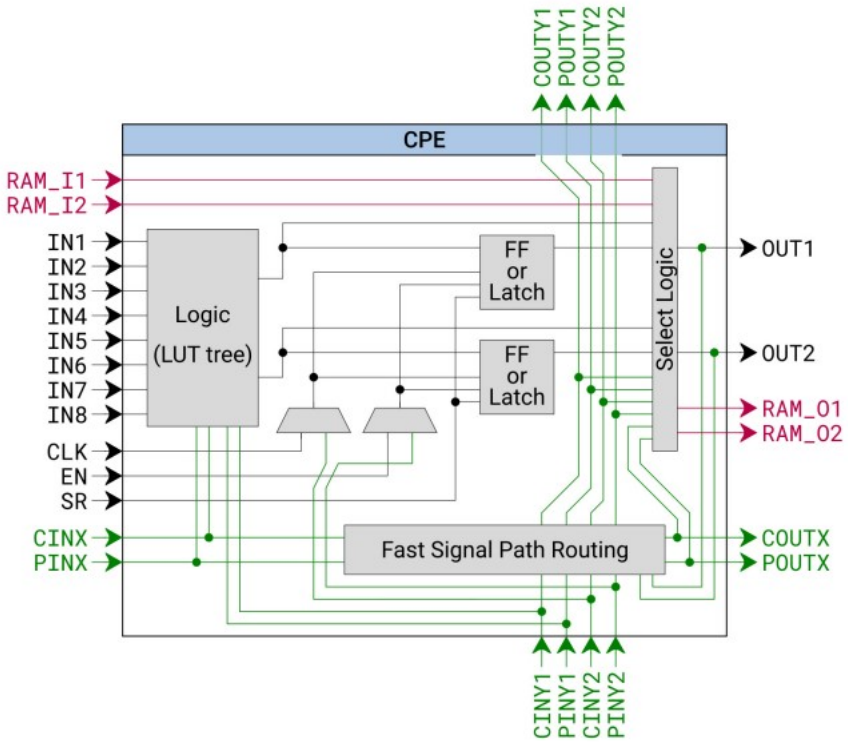


Figure 4.3: CPE logic element [8]

Positioned between the CPEs and the routing structure are the embedded memory cells. This set of 32 Block RAM (BRAM) cells can be configured as either a single 40K or two 20K Dual Port SRAM (DPSRAM) cells. Figure 4.4 shows the structure of the BRAM positioned in the FPGA.

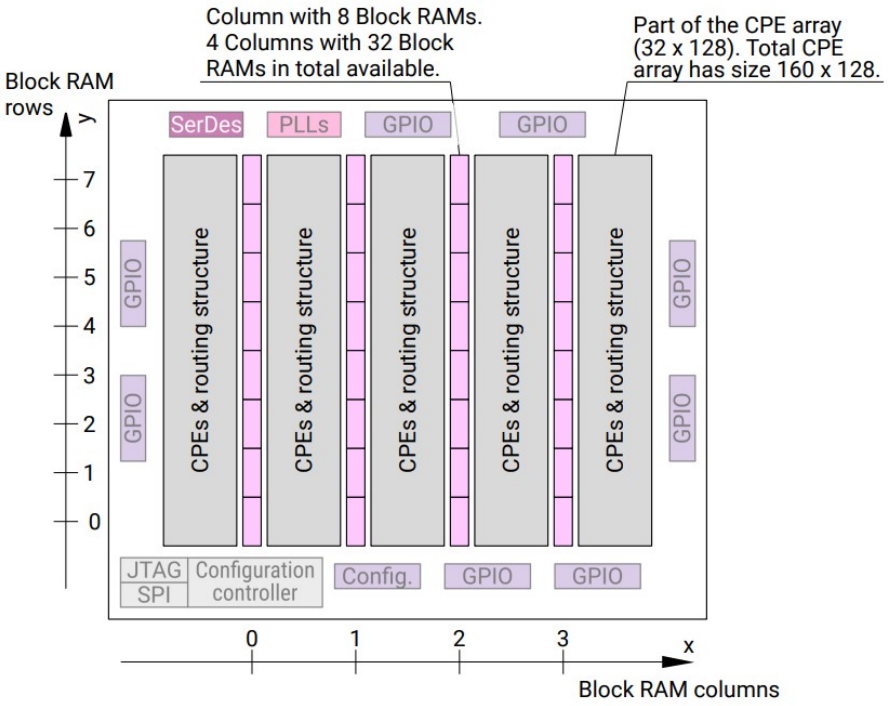


Figure 4.4: BRAM cells of the GateMate [8]

The GateMate's routing structure is of the island type. It consists of an array of 160 x 128 CPEs which are interconnected by 164 x 132 switch boxes of two different types. Small switch boxes can connect bi-directionally two other switch boxes both horizontally and vertically while big switch boxes can connect up to six other boxes in each of the specified directions. Both types of boxes offer the possibility of connecting to another box diagonal to the upper right and lower left direction. The layout is shown in figure 4.5.

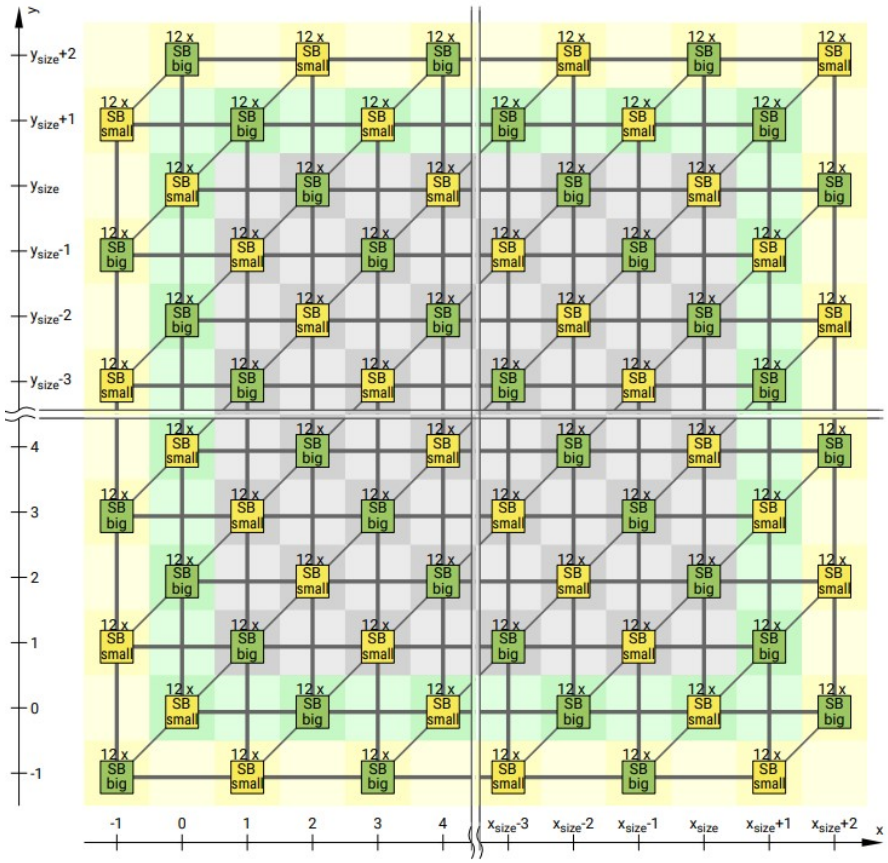


Figure 4.5: Routing structure of the GateMate [8]

The evaluation board has six available GPIO-Banks for connection of signals to the FPGA. Four of them can be set to a voltages of 1.2, 1.8 or 2.5 Volt while the other two are fixed to 2.5 Volt. The voltage setting can be configured by jumpers on the board. Five out of six GPIO connectors are utilized in this project along with a separately manufactured Printed Circuit Board (PCB) intended for the use of an FPGA Mezzanine Card (FMC) connector. The power supply for the GPIOs can be driven by the GateMate or from a connected device. Figure 4.6 shows the general overview of the pins for all GPIO-Banks.

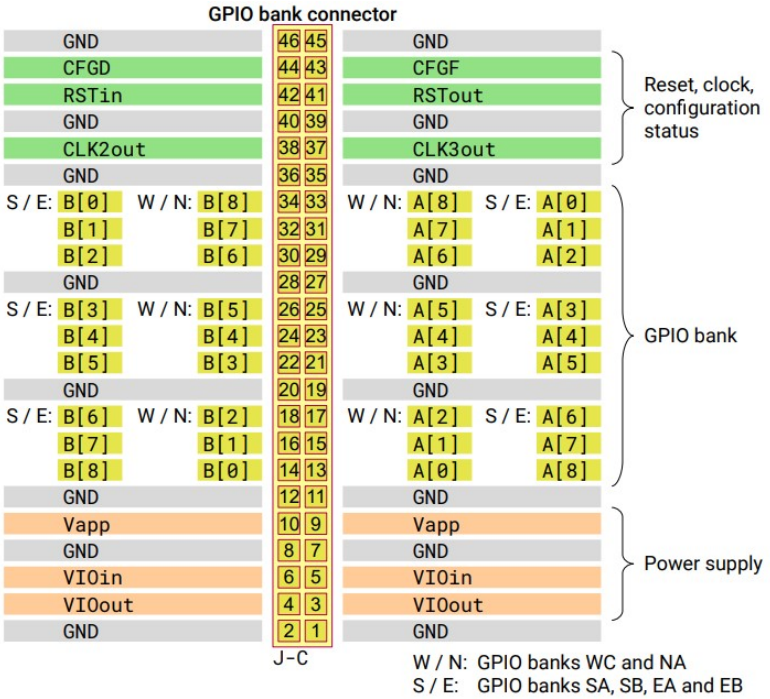


Figure 4.6: GPIO-Bank scheme on the evaluation board [9]

Furthermore the evaluation board contains two PMOD connectors, which adhere to an open standard for the connection of peripheral modules. The voltage can be set to 2.5 or 3.3 Volt for operation. Figure 4.7 shows the connectors on the evaluation board.

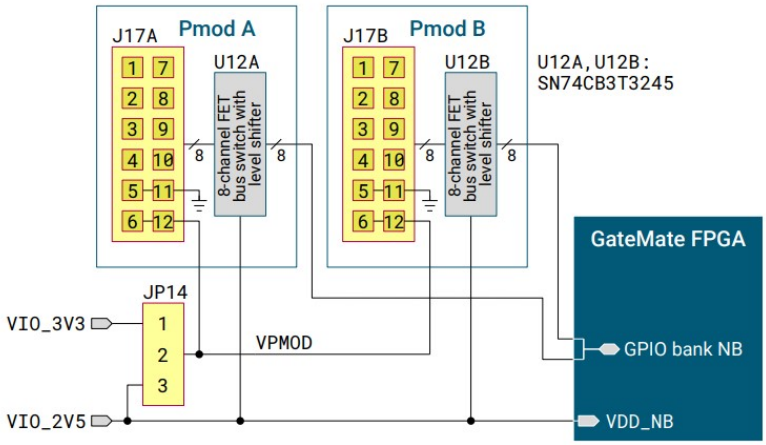


Figure 4.7: PMOD connectors of the evaluation board [9]

Programming the FPGA can be accomplished trough either the Joint Test Action Group (JTAG) or the Serial Peripheral Interface (SPI) connection, as shown in figure 4.8. The evaluation board is equipped with an Future Technology Devices International (FTDI) Universal Serial Bus (USB) to serial converter for this purpose. The programming mode setting can be adjusted using the switch SW1 located on the board. Furthermore the use of a flash memory to store the FPGA configuration is also possible, but was not used in this project.

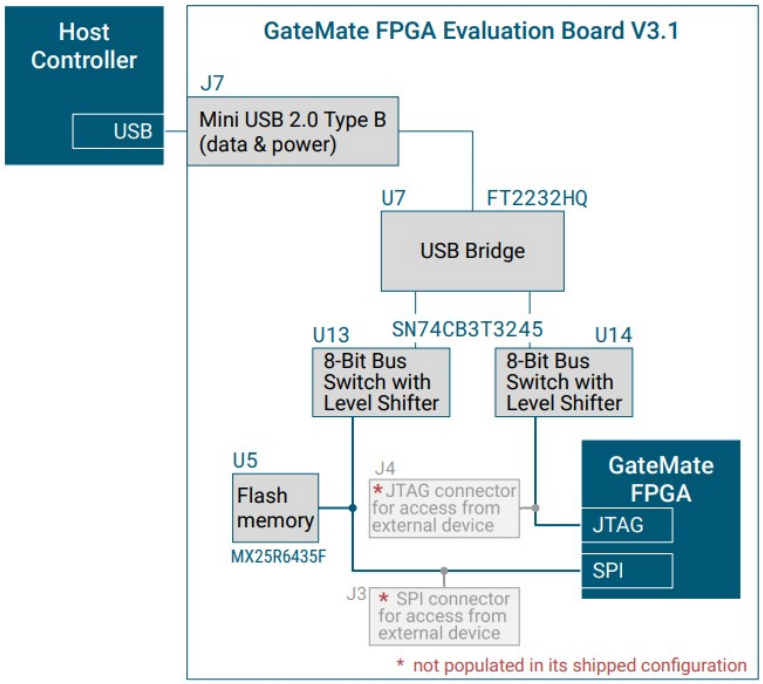


Figure 4.8: Setup of the JTAG and SPI connections [9]

The evaluation board requires a power supply of 5.0 Volt and has different converters on board to provide a wide voltage range for various applications. The supply voltage can be supplied via USB or by connection to the pins of J6 like shown in figure 4.9. Additionally, the device can be powered via the USB port used for programming or through a separate connector. The FPGA reset turns off only when all converters have reached a stable state after power-on. The FPGA core voltage can be adjusted in the range from 0.9 to 1.1 volts. The low power mode operates at 0.9V, the economy mode at 1.0V, and the speed mode at 1.1V, providing users with the flexibility to choose various power settings.

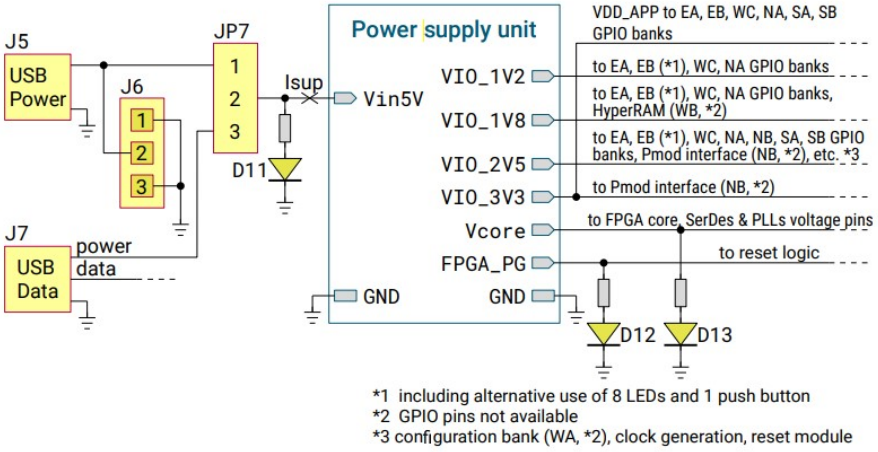


Figure 4.9: Power supply of the evaluation board [9]

After the user has developed a new hardware design for the GateMate FPGA, the subsequent stages involve synthesis, implementation and configuration. Figure 4.10 gives an overview of all necessary steps.

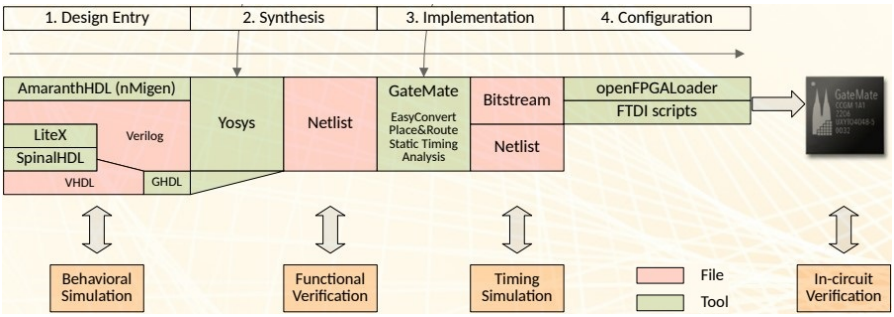


Figure 4.10: Design flow of the GateMate FPGA [8]

The design-file, which can be written in Very High Speed Integrated Circuits (VHSIC) Hardware Description Language (VHDL) or Verilog is then passed to the open-source synthesis framework Yosys[27]. Yosys accepts only Verilog files as input. In the case of a design in VHDL, the G Hardware Design Language (GHDL) VHDL analyser translates the VHDL design into a database that is usable by Yosys. The synthesis setup, generates a netlist file that can be processed by Cologne Chip's Place and Route tool. Furthermore, a constrains file is needed, to declare the input and output pins. The Place and Route tool then generates the bitstream for configuring the GateMate. To flash the bitstream into the device, the JTAG or SPI connection can be used.

The synthesis as well as the Place and Route tool are generating more files in the procedures than just the ones mentioned. These are just the most important ones for generating the bitstream. With the other files it is possible to check the implementation results in more detail or to use a simulation tool to check the functionality of the design after each step.

The explanation in this thesis are limited to the main functions of the evaluation board used in this project. For further information the documents from Cologne Chip [9] or the master thesis [14] can be consulted.

4.2 Lattice iCE40 UltraLite

The iCE40 UltraLite [30] is a low power SRAM FPGA manufactured in a 40nm CMOS low power process by Lattice [3]. This FPGA type has a huge device family with different versions of size and additional functions, with focus on a very small form factor. The FPGA used in this project is the iCE40UL-1K shown as a block diagram in figure 4.11.

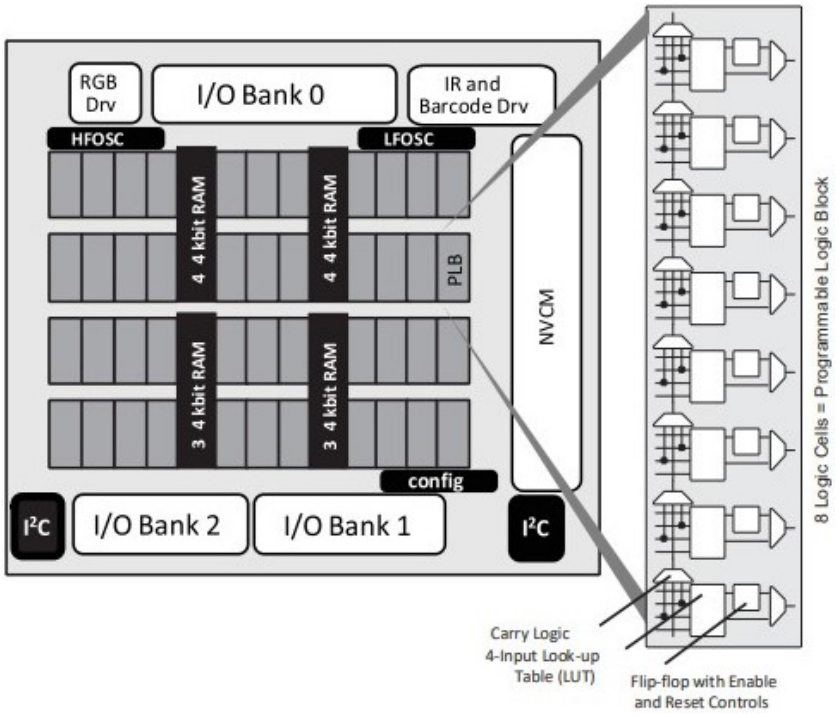


Figure 4.11: Block diagram of the iCE40 UltraLite FPGA [30]

Features of the FPGA are as follows,

- 1248 LUTs,
- Embedded Block RAM (EBR) of 56kbits,
- 10kHz and 48MHz oscillator,
- 1 clock generator (PLL),
- 26 I/O-Ports,
- 1 Non Volatile Configuration Memory (NVCM).

[3] `https://www.latticesemi.com/`

The iCE40 UltraLite is based on Programmable Logic Blocks (PLBs), which consist out of eight Logic Cells (LCs) with four input LUTss, a D-style Flip-Flop (DFF) and Carry Logic, as shown in figure 4.12. PLBs can be used for both logic and arithmetic functions, depending on the specific need.

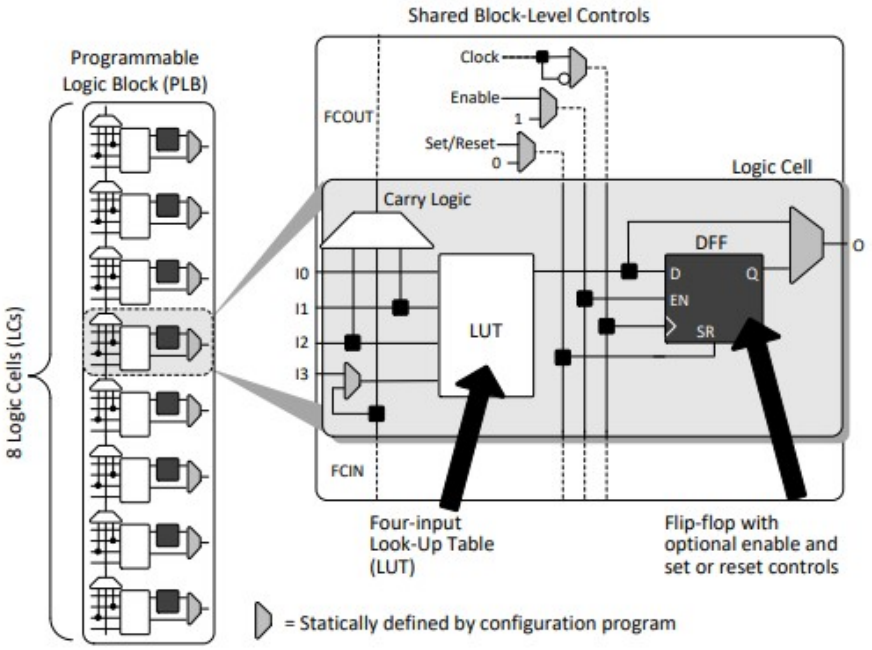


Figure 4.12: PLB of the iCE40 UltraLite [30]

The SRAM FPGA includes a Non Volatile Configuration Memory (NVCM) that can only be programmed once. If programmed, the SRAM will be automatically configured with this design on startup. However, it is still possible to configure the SRAM from an external programming device. If the NVCM is not programmed, the SRAM can be configured by an external flash or directly from an external device. The options for configuration can be seen in figure 4.13.

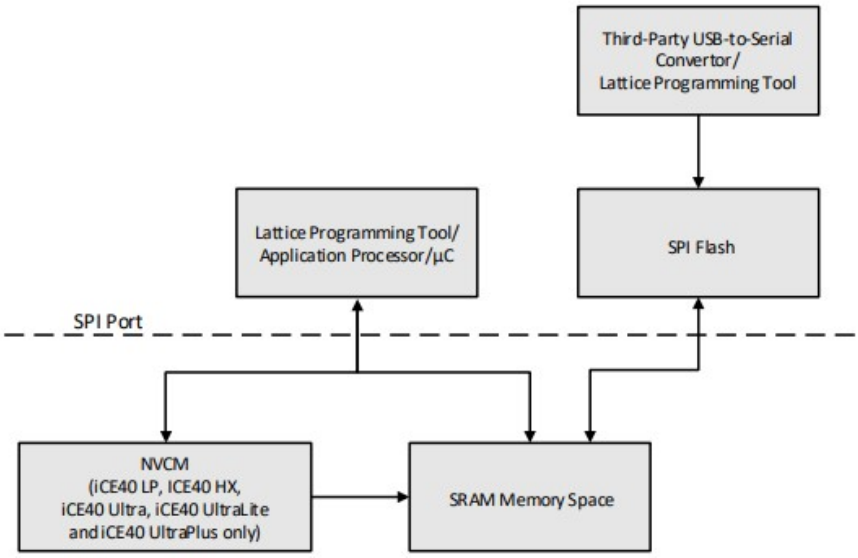


Figure 4.13: Configuration options for the iCE40 UltraLite [29]

The iCE40 UltraLite features on-chip Power-On Reset (POR) circuitry to guarantee a secure start-up once the power supply reaches a dedicated level. The POR circuitry monitors three voltage levels: the core supply, the NVCM supply and the I/O supply voltage. It triggers the configuration of the SRAM from the NVCM or the external flash memory.

Additionally, the device has 56 kbits of embedded block Random Access Memory (RAM), divided into 4 kbit RAM blocks between the PLBs. The three I/O banks can support 1.8, 2.5 and 3.3 volts and can be configured using the iCEcube2 software, which is described in chapter 5.

The iCE40UL is utilized on an evaluation board, known as the Breakout Board, which is shown in figure 4.14. This board is used for configuration of the FPGA from a flash memory and connecting it to the tester FPGA. The PMOD connectors are used for the connection between the two devices, as the test setup does not require many I/O pins. The Breakout Board provides a 12MHz clock for the USB interface which is also linked to the FPGA and serves as the system clock for the setup. Figure 4.14 displays the Breakout Board, with the FPGA itself indicated by a white arrow in the lower right area.

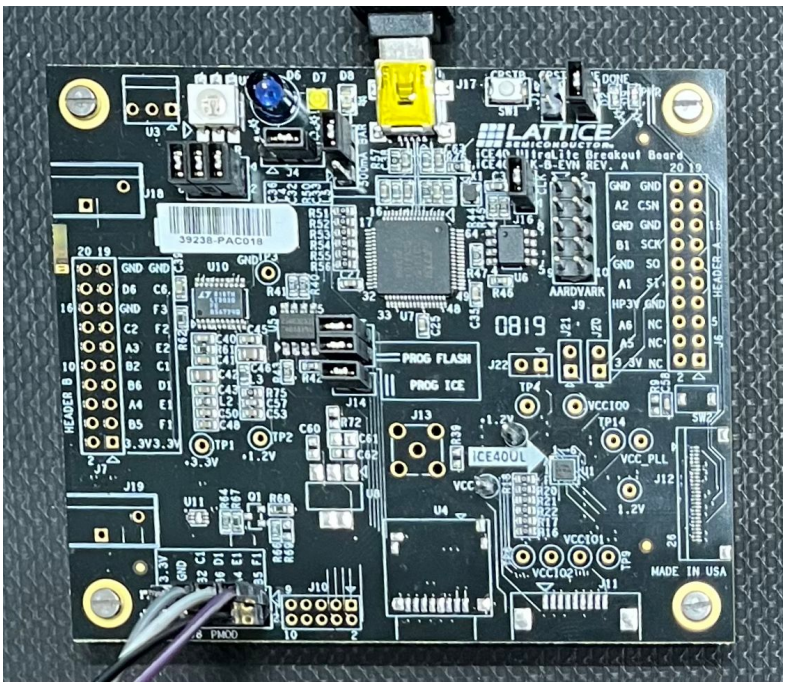


Figure 4.14: iCE40 UltraLite Breakout Board

The Breakout Board provides an FTDI chip to program the flash memory of the board or to directly configure the SRAM via a USB connection. The Diamond Programmer software can be used for configuration of the device, as described in chapter 5.

The iCE40 UltraLite offers additional features, such as direct high current drive outputs for Light Emitting Diodes (LEDs). This enables the direct driving of an RGB-LED, which is also positioned on the Breakout Board. This chapter provides a brief overview of only the relevant basics, as not all functionalities are used or of interest at this point of testing. For additional information, please refer to the documentation provided by Lattice [30].

5 Soft- and Hardware

The Appendix A includes a list of the used software and hardware in this project. Furthermore this section gives some short information referring to these, except of the GateMate evaluation board and the Lattice iCE40 UltraLite Breakout Board which are explained in the previous chapter 4.

5.1 Vivado and Vitis

Vivado and Vitis are software-tools from Xilinx for the implementation of hardware designs and programming of embedded ARM processors. With Vivado a hardware design can be written in a HDL language such as Verilog or VHDL. It also provides the ability to simulate the design before using it in real hardware. When the design is finished, it can be loaded into the FPGA and used with Vitis to program for example C-code for the ARM part of a Zynq device. The FPGAs used with this software are indeed Zynq devices integrated on a Zedboard and a MicroZedboard which are described in section 5.3 of this chapter.

5.2 iCE cube2 and Diamond Programmer

The iCE cube2 design software is a specialized tool for the ultra-low power devices from Lattice. It can synthesize and implement hardware designs for the iCE40 FPGA family and provide simulation files. To load the design into the FPGA or flash memory, the Diamond Programmer software from Lattice is required. Additionally, a Lattice programmer tool is required for programming the NVCM, although it was not used during this testing phase.

5.3 Zedboard and MicroZedboard

Both devices are evaluation boards manufactured by Avnet [1]. Main part is the Zynq SoC Chip by Xilinx which includes an FPGA and two ARM cores. The Zedboard [28] is used in this project to control the setup of the leakage current test by communicating with various devices. Additionally the B13 benchmark test is prepared with this FPGA. After

[1] `https://www.avnet.com/wps/portal/us`

configuring the FPGA design, PetaLinux is installed for execution on the ARM cores. This setup is then used with Vitis, to write C programs which are then executed on the ARM cores to control the tests. The advantage of this approach is that the system can be controlled from a standard terminal on a computer or via an Ethernet connection.
The MicroZedboard [17], in conjunction with an FMC carrier card, is utilized at CERN to control all test setups. The hardware designs are implemented on the FPGA, while the ARM cores run a PYNQ-Framework [20], which is based on a Linux image. This provides the capability to use JupyterLab [21] and write the controlling of the tests in Python code.

5.4 Thermostream

The Thermostream is a device that can provide a dry airstream with a temperature from -90°C up to 300°C. In addition, the temperature can be changed very quickly in small or large increments with extreme precision. For this reason it is used to test the FPGA in its possible temperature range. The Thermostream can be controlled to change the temperature in defined steps while the temperature on the device is measured with a sensor, to reach exact results.

5.5 DMM Measuring device and HMC Power supply

An HMC 8043 power supply is used to power the test-setup, which can provide different voltages or currents on up to three channels. By using the sense connection, the supplied voltage can be measured close to the device and the power supply can be controlled more precisely to the target value.
The Digital Multimeter (DMM) can measure voltage, current or ohmic resistance with high resolution. It can be set to the expected measurement range or to an automated mode to select the optimum depending on the input being measured.
Both instruments support the LAN eXtension for Instrumentation (LXI)-protocol [2] for connection to a control system. This allows data to be sent and read from a remote instrument.

6 Leakage current of the GateMate

The leakage current of an FPGA varies significantly with the temperature of the chip during operation, which also affects the total power dissipation depending on the specific operating point. Thus, the knowledge about an occurring leakage current is quite relevant to take countermeasures, if necessary. To differentiate between dynamic and leakage current consumption the chip is hold in reset. The GateMate FPGA is qualified for core voltages of 0.9V, 1.0V and 1.1V and for temperatures from -40°C to 125°C and is therefore also tested in this temperature and voltage range.
The measurements are done with two different types of FPGA chips. The first test is done with an evaluation board, where the FPGA is a so called short loop chip [13]. This type is processed in a shorter production than the normal one and therefore is not processed as detailed as in a full loop process [13]. The short loop process is more popular for new or faster needed devices and is also less expensive because of the smaller amount of time consuming production steps. The second test is done with a tester board on which the FPGA itself is mounted in a socket for easy replacement. Furthermore the FPGA chip comes from a full loop production.

6.1 Leakage current measurement setup

The test system makes use of the Thermostream climate device described in section 5.4 to set the temperature in the desired range without causing condensation problems at cold temperatures. An HMC 8043 power supply is used to power the GateMate, and a DMM 6500 multimeter is used to measure the leakage current. The Zedboard controls the measurement. It defines the temperature and voltage values and sends configuration commands to set these values at the devices. In addition it collects the measured data. This is accomplished with the LXI protocol using Ethernet based communication.
The basic control functions are pre-programmed and ready for use. These include the initialization of the communication stack and commands to set or request data. Communication with the power supply and the measuring device is done via the LXI-protocol. The Thermostream is controlled with specified commands via Ethernet, which are prepared as well.

Figure 6.1 displays the setup. The Zedboard is located on the right side, with the supply on top of the measuring device. The outlet of the Thermostream is on the left side, leading into a box where the GateMate is positioned. The air outlet is directed straight

at the FPGA chip.

Figure 6.1: Setup of the leakage current test

A temperature sensor is placed on top of the FPGA chip to ensure that the temperature of the device is measured more accurately. Figure 6.2 shows the setup inside of the box, with the sensor in the middle of the chip.

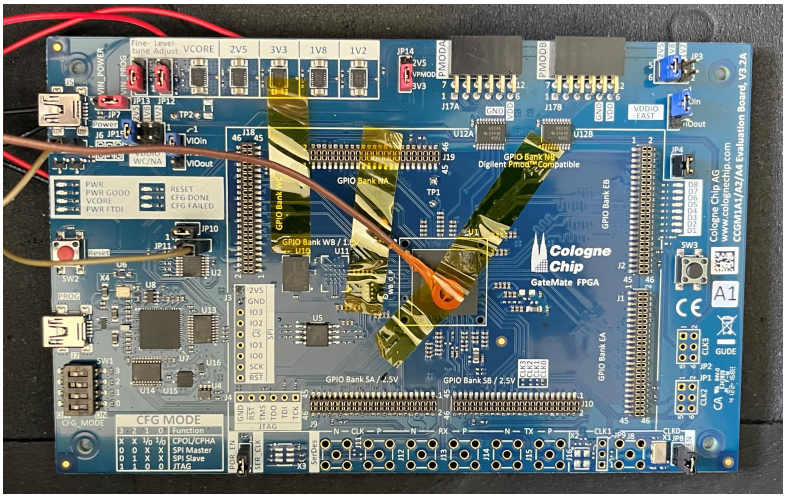


Figure 6.2: Temperature sensor on the GateMate FPGA

The core of the GateMate is supplied separately from the rest of the board.This is done by removing the zero ohm resistor (R86) under the VCORE power converter. Its position can be seen in the figure 6.3 in the upper area in the red circle to the right of the test point TP2. The power supply in line with the measuring device is connected to the test point TP2. The actual voltage closer to the core is measured at the test point TP1 with the sense connection of the power supply, to ensure that the required voltage reaches the core. The circle on the right of the figure 6.3 shows the test point TP1. The rest of the board is supplied with 2.5 volts to power the POR of the GateMate. Using the POR module with the reset signal on pin 3 of JP11 connected to the ground pin on J6 provides a safe reset of the board for the entire test. The red arrow in figure 6.3 points to the two connected pins by the brown wire.

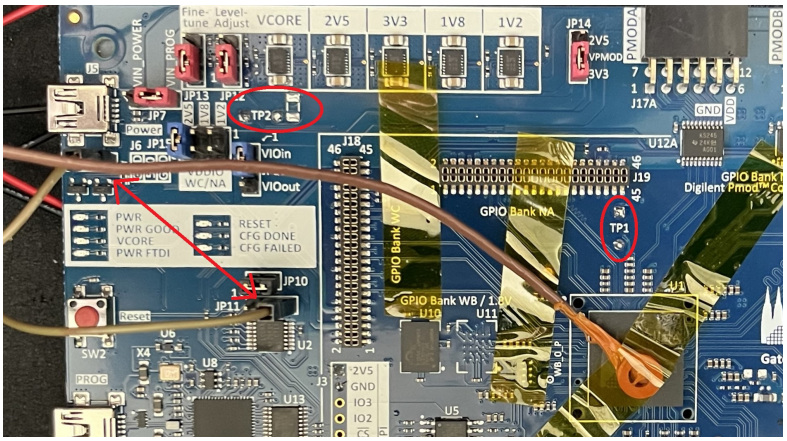


Figure 6.3: Connection points on the GateMate evaluation board

6.2 Programming in Vitis and test-procedure

A basic FPGA hardware design is prepared for the Zedboard, which is then used in combination with a PetaLinux installation running on one of the ARM cores of the Zynq device. On this platform the software is programmed in Vitis in C. A tutorial explaining how to use this platform with Vitis can be found in Appendix C of this thesis.

A procedure is programmed to control all settings and to perform a complete test. This program contains the basic values for this test setup as initial parameters. These include the start and end parameters for the temperature and core voltage, together with the step size. The number of measurement points to be taken at each set point with a user-definable waiting time is also preset.

When the program is started, the user can specify each previous named parameter for

the test, or the default parameters can be left preset. All actual values set for the test are then displayed and the user is prompted to confirm the setup to begin the test. When this is done, the Thermostream begins to cool down the air-temperature to the specified minimum value. When the temperature is reached, a user-definable time must be waited to ensure that the DUT is also at the correct temperature. The test starts by measuring the leakage current and then adjusting the three voltage settings for the core voltage. When the leakage current is measured for all the three voltages, the next temperature is set and the process begins again until the whole temperature range is covered. During the test, the measured values are stored in a file on the SD card of the Zedboard. This file can be extracted afterwards to evaluate the data from the test. When the maximum temperature is reached, the measurement is stopped and the device is cooled down to room temperature.
Figure 6.4 shows the results of the first test with the GateMate evaluation board and the short loop chip.

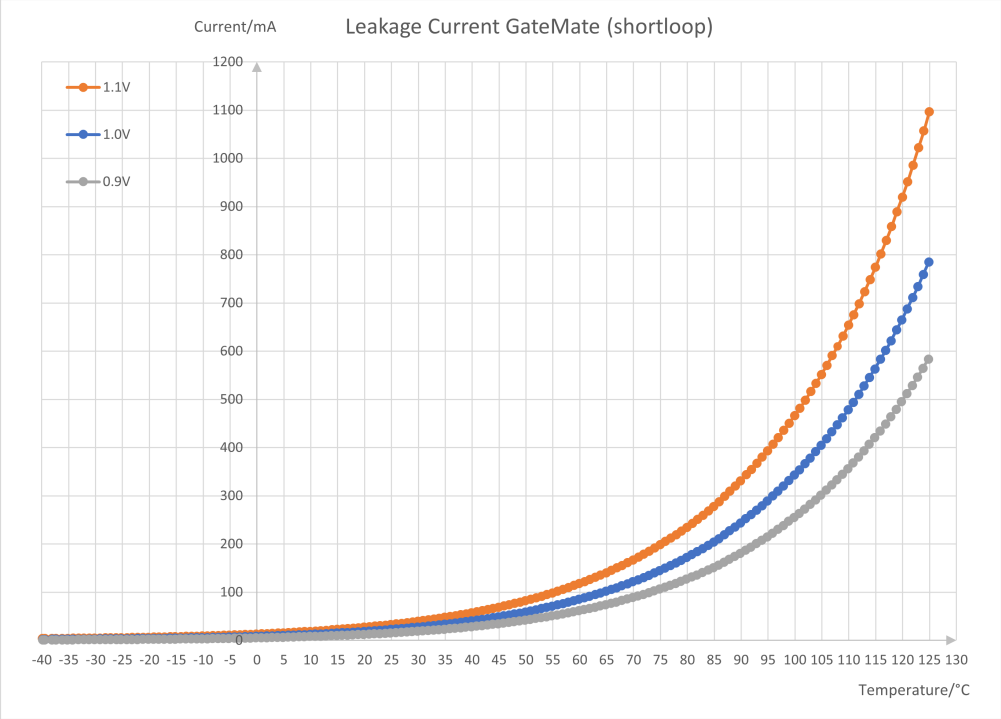


Figure 6.4: Resulting graph for leakage current of the short loop chip.

The leakage current increases with the temperature exponentially and the measured values show no unexpected outliers. The whole test took about four hours, including the cool down procedure at the beginning and the cool down step to room temperature

at the end.

For the next test, with the second type of FPGA chip measurement step size is increased to five degrees. In addition, the waiting time after reaching each temperature value is increased and more measurement points per set point are taken. With more values the average can be calculated more accurately and it can be checked if there is a difference between the first and the last measured current value. If a difference is detected, this indicates that the FPGA has not reached the target temperature and a longer waiting time is required. The results of this test can be seen in the following figure 6.5.

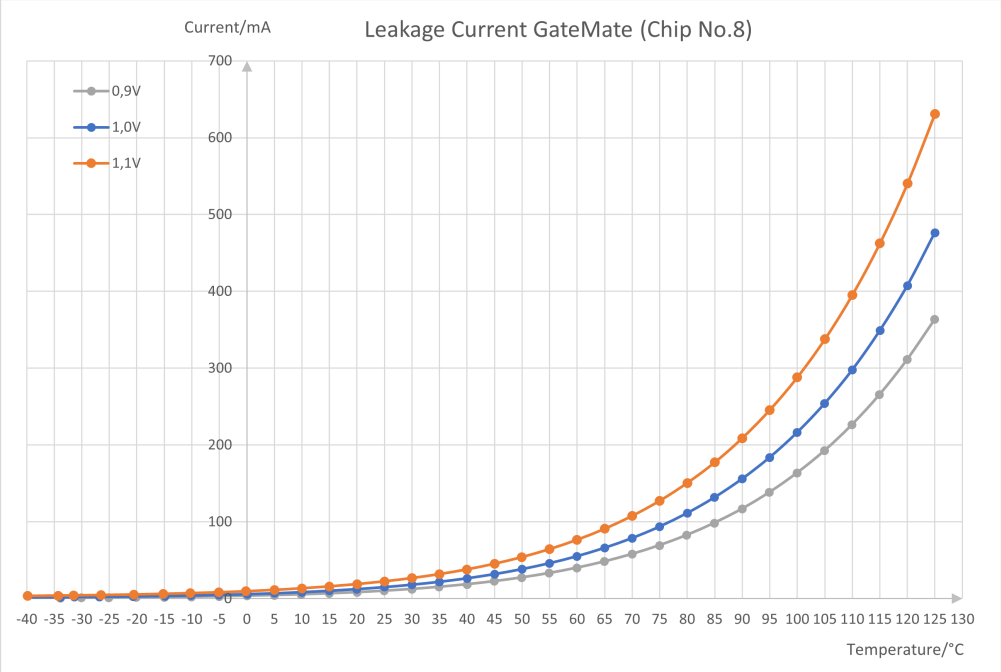


Figure 6.5: Resulting graph for leakage current of the full loop FPGA type.

It shows, that the FPGA of the second charge has a significantly lower leakage than the first. To prove this behaviour, this test is done with three FPGA chips from the second type. On the evaluation board with the socket, these chips can be easily replaced and tested in the same setup. The leakage of all three chips show the same behaviour with comparable values in all ranges. Furthermore the results for both types confirm the expected exponential increase in leakage current as a function of the temperature, as explained in chapter 3

7 Radiation qualification tests

The qualification tests at CERN focus on different parts of an FPGA to determine their sensitivity. For the reason that this is giving just partial informations, an application test is necessary to give an example for a real world application. Furthermore this application test is standardised and comparable between different devices.

In preparation for the tests at CERN, the previously performed tests of the GateMate [14] are explained in their functionality. The B13 benchmark test was then prepared on the GateMate. Due to the fact, that the Fachhochschule Dortmund (FHDO) does not have a MicroZedboard with an installation of a Pynq environment, which is the setup for the testing at CERN, all preparation was done with a Zedboard. By using a basic hardware design in a PetaLinux installation, the kernel was prepared for the use with Vitis. This setup provides a comparable system environment to the one that is used at CERN and was therefore a good starting point. At CERN the setup has been transferred to the environment with the MicroZedboard for testing. Furthermore a test setup to determine the cross section of the GateMates Configuration RAM (CRAM) has been implemented and tested.
The last test setup was done with the Lattice iCE40 UltraLite. An application test has been implemented to obtain initial results and determine if further investigations on this device could be of interest.

7.1 System setup

As mentioned above the system used for preparation at FH Dortmund is based on a Zedboard. The same FPGA is used on the Zedboard as on the MicroZed, which is why the hardware design can be used with only a few adjustments. The main difference between the test setup at CERN and at Dortmund University of Applied Sciences and Arts is the programming of the control system. The Zedboard is programmed with Vitis in C code, while the MicroZed Board can be programmed in a Pynq environment with Jupyterlab in Python. The setup with the Zedboard is only used for the benchmark test on the GateMate. The other tests are directly done at CERN with the MicroZed Board as tester device.

To connect the tester and the DUT, the FMC cable and the already mentioned PCB-adapter for the GateMate are used. For the test setup of the iCE40 UltaLite a PMOD

connector is used. Only a few signals are needed for the watchdog setup and another PCB-adapter for the FMC cable is not necessary at this point of testing.

7.2 Testing of the GateMate

Figure 7.1 shows the system setup for all tests with the MicroZedboard as tester and the GateMate as DUT.

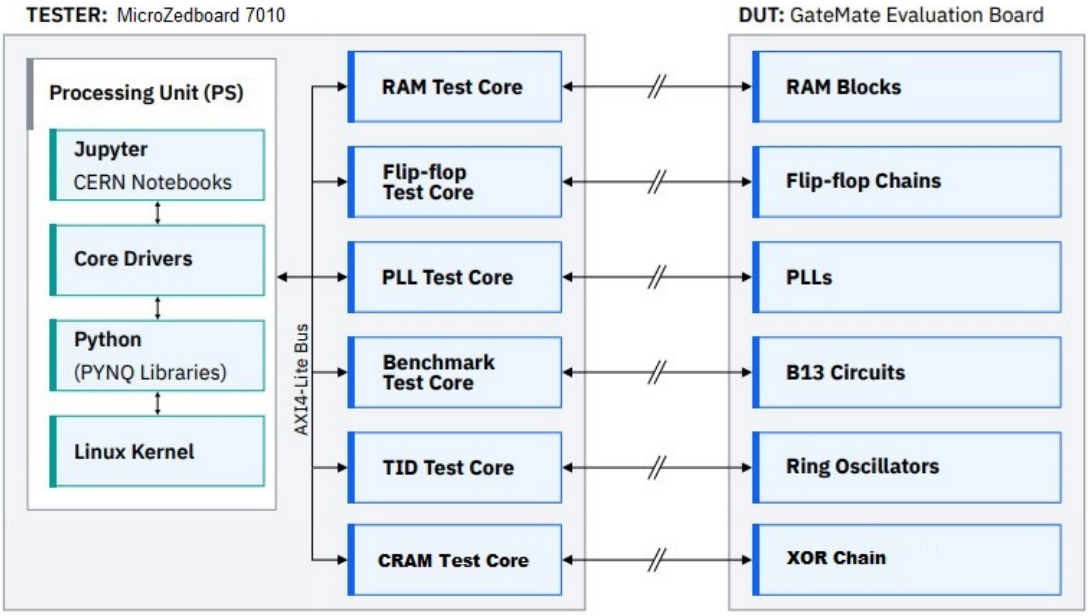


Figure 7.1: Overview of the general test-setup [14].

7.2.1 Phase Locked Loop (PLL)

The GateMate has four PLL's included in the FPGA which can be used to generate different clock frequencies if needed. Each one of the PLL's is fed by a clock signal from the tester board. The PLL has a lock signal to verify that the output is at the right frequency. This lock signal is sent back to the tester and checked for changes while the FPGA is being irradiated. The setup of the system is shown in the figure 7.2.

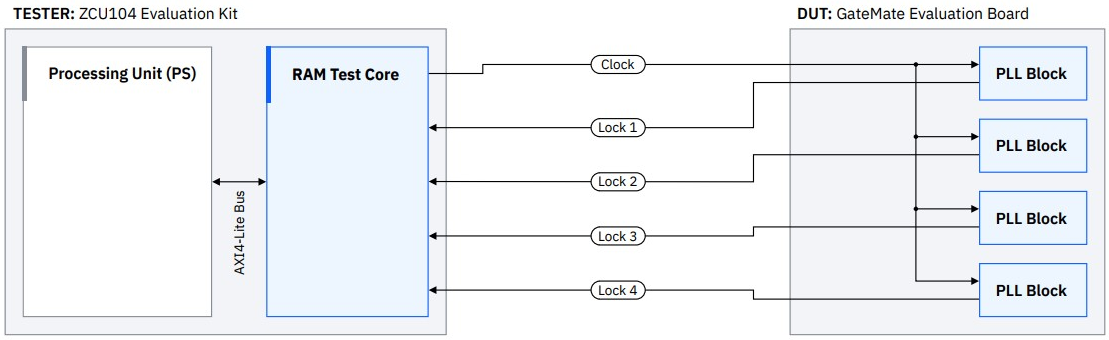


Figure 7.2: Setup for the PLL test[14].

The IP-core of the tester is taken from [23] and adapted for the GateMate by [14]. It transmits a clock signal and receives the lock signal of each PLL from the DUT while the system is irradiated. Whenever the lock signal is lost, this is a indication that a PLL has been affected by the radiation. Each loss is counted by the tester.

7.2.2 Flip-flops

In this configuration, a specific number of flip-flops are connected in a series to a shift register. The first one has a defined input and every subsequent one has the output of the previous one as input. A Window Shift Register (WSR) is used to avoid integrity problems that may arise at higher frequencies when a full shift register is read out in parallel. With this type of shift register, only a limited number of flip-flops are read out and displayed in a so called window. The window loading signal is connected to the clock input of these flip-flops to update them. An example of a short flip-flop chain can be seen in figure 7.3.

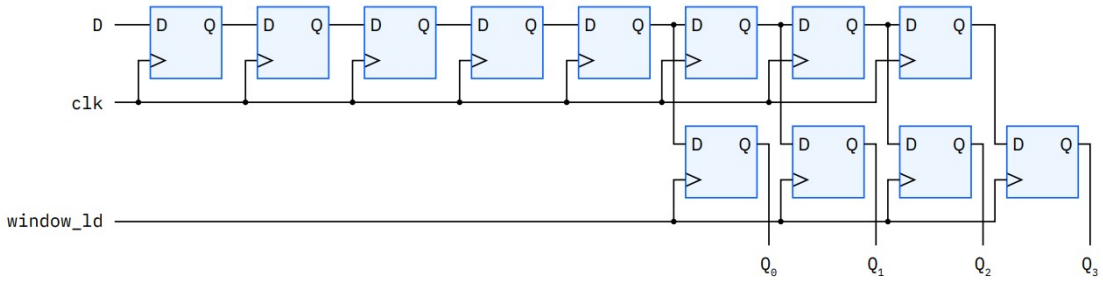


Figure 7.3: Chain on flip-flops for the test [14].

The IP-core used in this test was originally taken from [23] and is already prepared for the GateMate in [14]. It provides the DUT with a clock, a window load and a pattern

signal. The signals from the DUT are the output windows of each WSR. The principle of this test setup is shown in figure 7.4.

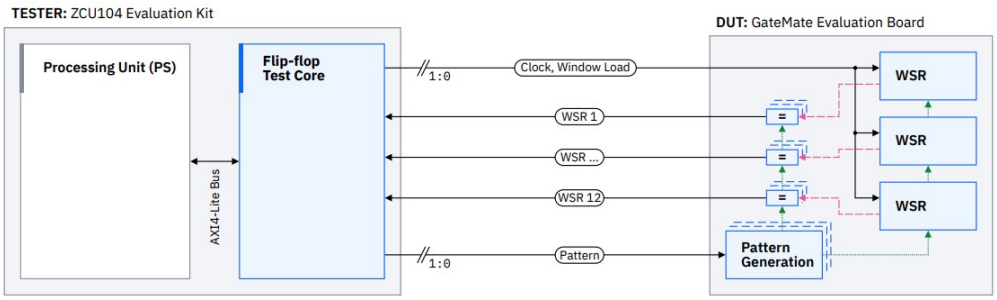


Figure 7.4: Setup of the flip-flop test [14].

One test method applies a static 0 or 1 to the shift register input and checks the output window for a change that is the result of a bit flip. A second method is to input an alternating signal that changes between 0 and 1 every clock cycle. For this method, the window load signal has to be pulsed with a specific frequency, to get a static output. When the output changes, a bit flip has occurred. In this case a bit flip is passed trough the flip-flops until it is processed by the next window load signal, as can be seen in figure 7.5.

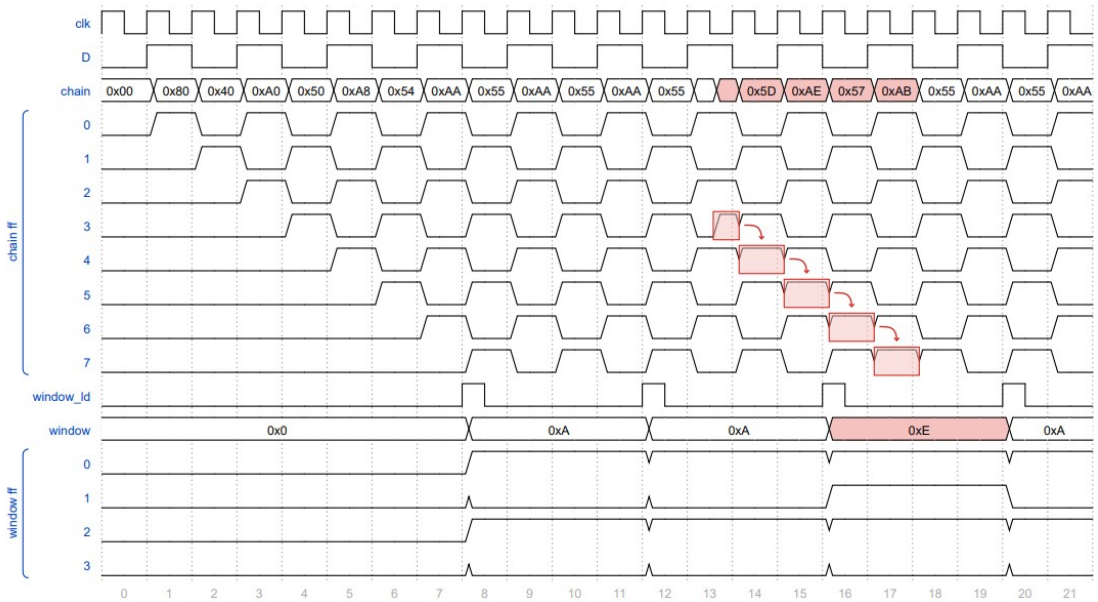


Figure 7.5: Processing of a bitflip in the flip-flop design [14].

The frequency for the load signal can be calculated by the equation 7.1.

$$f_{window} = \frac{f_{clk}}{n} \tag{7.1}$$

7.2.3 BRAM cells

For this test, data is written to the block memories of the DUT before the test begins. During irradiation, the data is periodically read out to detect any changes in values. If one or more memory bits are affected, SEU's or MEU's may have occurred. In this case, the test is halted and then restarted once the initial state has been rewritten to the memory. The system setup is shown in figure 7.6.

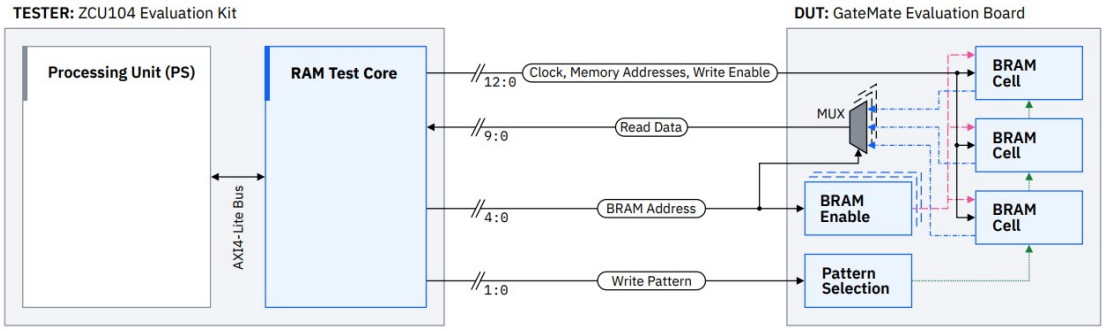


Figure 7.6: Setup of the memory test [14].

The IP core of the tester sends the clock signal and a signal to select the initial data to write. To write to a specific memory block an address which identifies the according BRAM, as well as the address of the word to be written and a write enable signal is sent. The memory data selected in three ways trough the "Write Pattern" signal. Available data options are all zeros, all ones, or a toggle between 0 and 1 for each bit position. The requested memory data is sent back by the DUT. The IP core for this system has already been prepared and tested [14] for the GateMate.

7.2.4 Total Ionizing Dose (TID)

The aim of the TID test is to measure the difference in propagation delay of a circuit inside the DUT induced by the accumulated radiation dose. This is achieved using ring oscillators, which consist of a chain of an odd number of inverters. The output of the last inverter is fed back to the input of the first inverter. The ring oscillator frequency reduces with increasing propagation delay of the elements within the circuit. By measuring the

DUT output frequency, it is possible to observe how strongly the absorbed radiation dose affects the FPGA. The principle of the test is shown in figure 7.7.

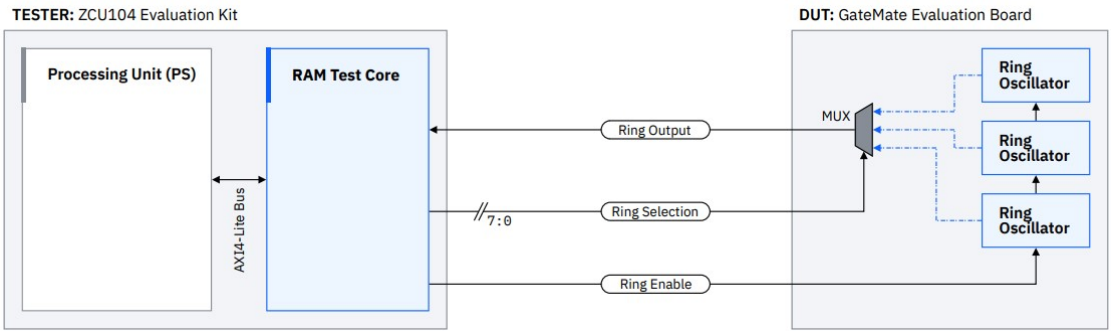


Figure 7.7: Design of the TID test [14].

The IP-core is prepared by [14] and was originally taken from [23]. By sending the ring selection signal, the desired output of a chain is selected and returned by the DUT. With the enable signal, the outputs of the elements in the chain are enabled and the oscillator starts oscillating.
Since the GateMate implementation software optimizes away series-connected inverters, the chain is built with D-latches instead. By inverting the last output and feeding back to the input of the first latch, a periodic signal is generated. This principle is shown in figure 7.8.

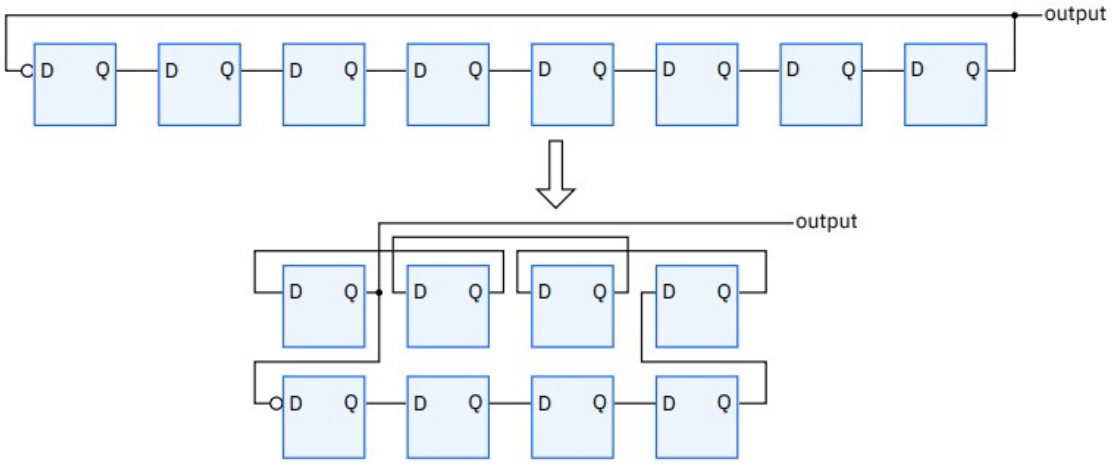


Figure 7.8: Example for chain of D-latches as used in the GateMate design [14].

The expected output frequency of the DUT can be calculated by equation 7.2. The number of inverters is denoted by n and the runtime of each inverter by t.

$$f = \frac{1}{2 * n * t} \tag{7.2}$$

The GateMate Place and Route tool can place the D-latches of a connected chain at irregular distance from each other, which can lead to different delay times and thus oscillation frequencies. To prevent this from happening, the position of each CPE in which D-Latches are used is defined manually for all chains. A Python script is used to generate a list of CPE positions. This script is already written and details can be found in [14]. The generated list is written in the ccf-file configuration file of the Place and Route software. The tool utilizes these positions to establish a consistent and uniform configuration for each chain, resulting in an equal delay for all paths.

7.2.4.1 TID with focus on ring oscillator frequency

The setup of ring oscillators in this test can also be used to create a Physically Unclonable Function (PUF) [19]. These structures have been tested in different setups in combination with the GateMate at the FH Dortmund but not under radiation. A PUF generates a unique response that cannot be replicated by another device. The frequency output of ring oscillators, which is unique due to process variations in production, can be used for the implementation of a PUF. Thus, the frequency of the ring oscillators implemented in the TID test is also unique. This feature is suitable for safety functions as it cannot be replicated by other devices. For a perfect setup, the oscillator frequency should be always reproducible. However, various factors such as temperature or voltage supply variation can affect the oscillator frequency. In this setup, the impact of radiation should be examined. The frequency output of the ring oscillators will be compared to each other. To obtain a reproducible PUF response, the relation between two oscillators should remain consistent. Therefore, the frequency comparison should always stay either positive or negative. If the frequency of one oscillator changes more than the frequency of another and the relation changes from positive to negative or vice versa, their output will not be suitable for a PUF-based function.

During the TID test, done in 2023 [14], the frequency of all ring oscillators was documented under radiation for a total dose of up to 9kGy. This data can be analyzed for the suitability as a PUF primitive based on ring oscillators.

To determine if the ring oscillators can be used for this purpose, the frequency of each instantiated ring oscillator is compared to one another and checked for a change in sign. By subtracting the frequency of each ring oscillator from a chosen reference oscillator a positive or negative relation can be determined. This process is repeated for each dose level and the resulting values are compared in two ways. Firstly, based on the starting relation at 0Gy, and secondly, for each dose step and the following result. The results

indicate the number of changing relations in comparison to 0Gy and from one dose level to the next.

7.2.5 Benchmark B13

The benchmark test is taken from the ITC'99 suite which is developed by Politecnico di Torino[10]. The B13 design was originally an interface to a weather sensor and is now used for FPGA testing to study the impact of radiation on a design with realistic structure and complexity while allowing comparability with test results from different FPGA types and vendors. During this test several B13 module written in VHDL are instantiated and and implemented in the FPGA. If an SEU occurs while the system is running, the SEU is counted and the test is stopped and restarted. At the end of teste the total number of SEU's over a defined time is captured. Since the occurrence of SEFI's is expected it has to be checked if the device stopped working to assure continuation of the test by reprogramming.

The test is controlled by the code running on the Processing Unit of the tester FPGA which is either written in C and programmed by Vitis in case of the Zedboard or written in Python and programmed via Jupyterlab. By use of defined commands, data is written to registers to start and stop the test or to read results. A pattern signal is generated and fed into each B13 circuit. Depending on the input and the clock, the output changes in different intervals. The tester and the DUT receive the same input and in consequence the output should be the same as well. To prove this, both values are compared on the DUT. Since the output is not changed in every clock cycle, it is possible that a corrupted value is overwritten by a new one before it is processed further.

To run the benchmark test, the IP core of the tester has to provide several signals and save data in case of an occurring SEU. Figure 7.9 shows the principle of the benchmark design.

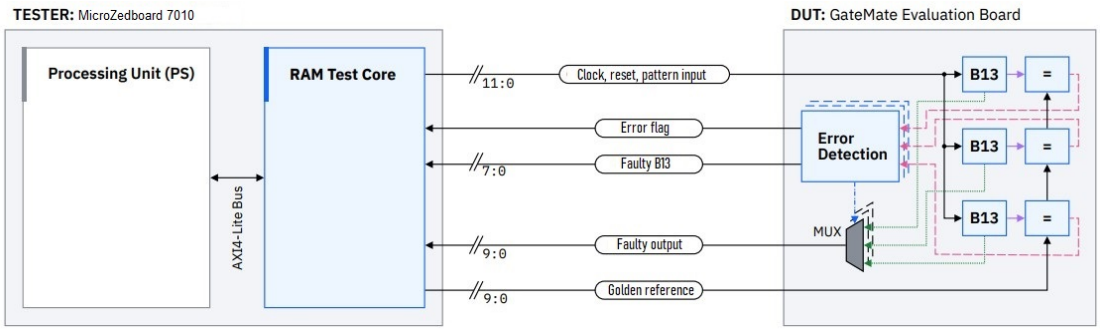


Figure 7.9: Design of the benchmark B13 test [14].

The tester transmits a clock signal to run the test. The DUT is provided with start and stop information through a reset signal. The pattern signal is given to each B13 circuit that is instantiated, to determine the output. The golden reference is then used, to compare the results on the DUT itself. The Device Under Test (DUT) will generate an error signal if the output of any B13 instance differs from the golden reference. The signal "faulty B13" indicates the number of the faulty B13 instance, while the faulty value is provided in the faulty output. If no errors are detected, the address and value of the first B13 instance are transmitted but not processed further. The data processed by the DUT is only stored on the tester device when the error signal is set high. The test is then reset and restarted.
When the First In First Out (FIFO)-Memory inside the tester is full, the test is stopped completely because no more data can be saved. The FIFO state is regularly checked, and new values are saved to a file to prevent memory overflow after several values have been stored. The FIFO can only be completely filled in the event of a SEFI. In this case, the testing process is halted and the GateMate is reprogrammed to resolve the error state.

7.2.6 Configuration Memory

Measuring the sensitivity of the CRAM is a crucial metric during qualification. The cross section can be calculated using the number of critical CRAM bits. Typically, this is achieved by dumping the CRAM before and after testing and counting the bitflips. However, the GateMates CRAM consist out of D-latches and does not provide the feature of dumping. Therefore, an alternative approach is used to determine the cross section. A design is implemented for which the number of critical bits is known. The design is based on XOR whose truth table is shown in table 7.1. The change of one bit lead to a change of the output.

Input 1	**Input 2**	**Output**
0	0	0
0	1	1
1	0	1
1	1	0

Table 7.1: Truth table of XOR logic.

The design interconnects all eight inputs of a CPEs LUT in an XOR structure to one output and repeats this multiple times in a chain. The first XOR in this design receives all inputs as zero. The output is then fed to one input of the next XOR, while the other seven inputs are provided with the same zero input as the first one. If one bit in this structure flips, the output will change as well and an SEU can be detected. Figure 7.10 illustrates the basic idea of this setup.

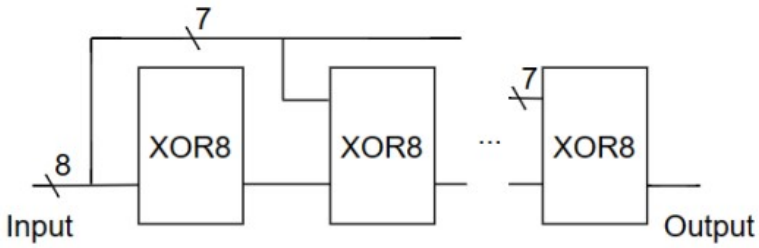


Figure 7.10: Principle of the CRAM test.

By setting all inputs to high or all to low a change of one bit will change the output. In this way a long chain of CPEs is implemented which are positioned manually, to control the setup and routing of the logic in the design. The positioning of the CPEs is done by a Python script that is adapted for this setup from the script used for the TID test by [14]. The chain is optimized for short connections between each CPE. This setup allows for the determination of the number of critical bits in the design. With this structure a chain of 13866 CPEs is implemented. During the test the last output is checked for any changes. The principle of the test design using the MicroZedboard and GateMate can be seen in figure 7.11.

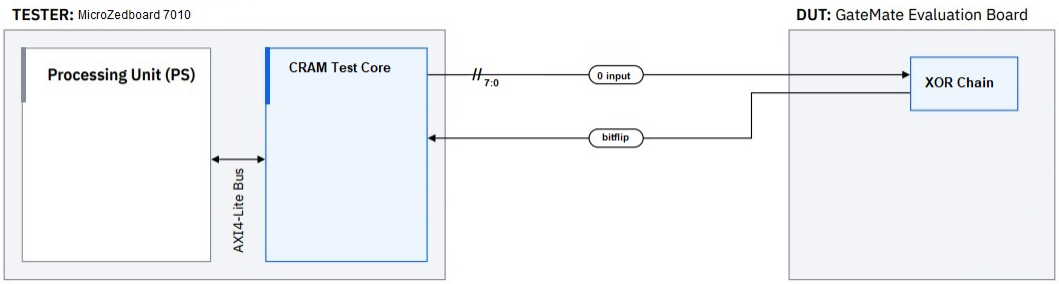


Figure 7.11: Design of the CRAM test.

The design input is static. As a result also the output should be static in SEU free operation. When ever the output changes, the design is known to be corrupted and a failure is counted. The test is stopped and, after the design is reconfigured, restarted. The time for a bitflip to propagate through the chain is measured before testing. This is necessary to ensure that the particle flux is set low enough so that, in case of an SEU, the event is counted before another particle can affect the output again.

7.3 Testing of the iCE40 UltraLite

The iCE40 UltraLite FPGA may be suitable for use in a specific project at CERN. To initially assess its potential for further investigation, a test setup is implemented using the iCE40 UltraLite as a watchdog. This application test is chosen for the reason that a similar functionality is of interest for use. If the test shows good results the qualification tests, done with the GateMate, would be the next step in the future.

7.3.1 Watchdog test

A watchdog in software is a function that checks a system for failures by sending a signal or pulse (wake) at dedicated intervals and receiving a response (done) back to confirm that the other system is still functioning. If no response is received, the watchdog resets the system.

The iCE40 UltraLite acts as a watchdog in this setup, sending a wake signal to a connected device, which is the MicroZedboard. The MicroZedboard must respond within a defined time frame with a done pulse to indicate that it is awake. If the done pulse is not received, the watchdog resets the other device. Thus, the watchdog sends two signals and receives one in return. The MicroZedboard acts as the connected device to be controlled by the watchdog and to check if the watchdog design is failing under radiation. If the wake signal is not received, the iCE40 UltraLite will be reconfigured. To test the reset function of the watchdog, the tester occasionally does not respond with the done pulse from time to time. If the watchdog fails to reset, it will also be reconfigured. Any reconfiguration will be considered as failure in calculating the cross section in the end. An example of the signals for this test can be seen in figure 7.12.

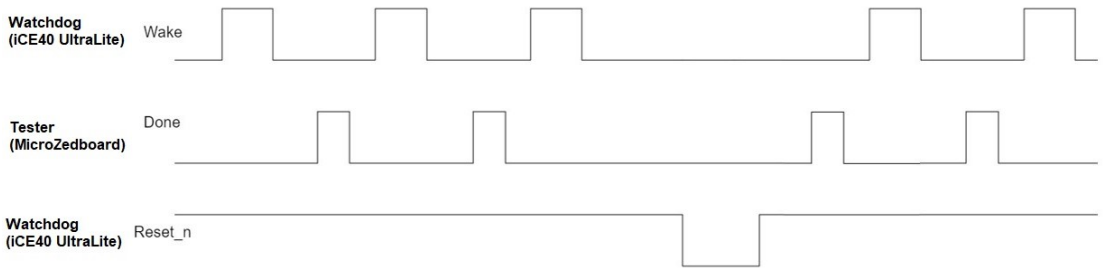


Figure 7.12: Principle of the Watchdog test.

The setup of the watchdog test with the MicroZedboard as the tester can be seen in figure 7.13.

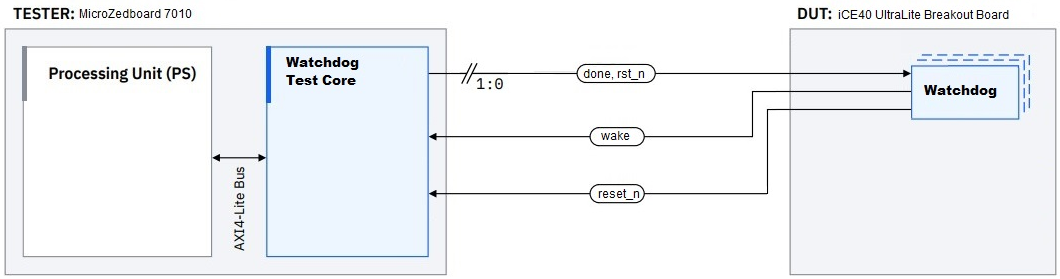


Figure 7.13: Setup of the Watchdog test.

The IP-Core for this test is newly designed to send and receive two signals. Three signals are used for the watchdog setup as previously explained. The „rst_n“ signal is used to reset the watchdog before starting the test. The system clock for each board is derived from its internal clock generator. The IP-Core of the MicroZedboard is clocked at 100MHz, while the Breakout Board provides the watchdog with a 12MHz clock. Therefore the timing for the signals is defined to ensure correct signal checking. The timing between two wake pulses is set to 100ms, while a wake pulse lasts for 20ms. A done pulse lasts 15ms, and the reset signal is held for 30ms. This timing reflects a possible use case.

8 Results of the qualification tests

This chapter summarizes the results achieved from three campaigns conducted at Paul Scherrer Institute (PSI) between September 2023 and November 2023. The radiation was administered using a 200MeV proton beam at the Proton Irradiation Facility (PIF) [12], which was accelerated by the COmpact MEdical Therapy cyclotron (COMET). The beam flux can be controlled by varying the current between 0.1nA and 10nA. The beam shape and size can be adjusted using copper collimators, after which the DUT is placed.

8.1 GateMate

The results of the B13 test and the CRAM test derived from the GateMate are presented. The achieved results from the previous campaigns can be found in [14]. In addition, the results from the TID test are evaluated again with focus on the relation of the frequencies between the ring oscillators. Figure 8.1 shows the GateMate installed in the irradiation facility at PSI.

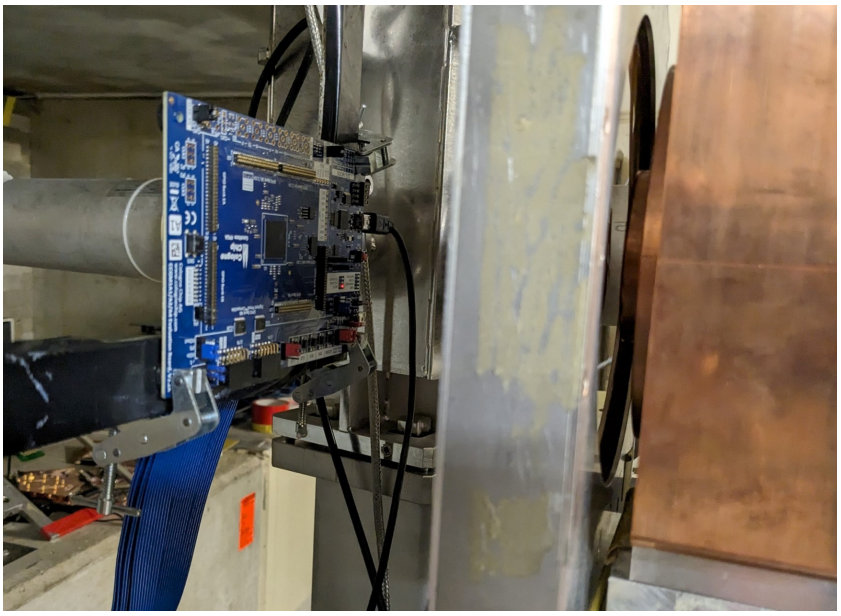


Figure 8.1: Picture of the GateMate at PSI.

8.1.1 B13 results

In the preparation setup with the Zedboard the benchmark B13 test could be implemented on the GateMate with close to 70% of the available CPEs. However, due to timing issues in the MicroZedboard setup, the exact same setup could not be reproduced at CERN. As a result, the B13 benchmark test could be implemented using approximately 60% of the CPEs in the GateMate. The design was implemented in a standard version with 100 B13 circuits and a triplicated version with a similar usage of the CPEs and 33 B13 circuits. Additionally, the design was tested prior to the campaign by injecting a false value to verify its correct processing, which was successful. The results of the tests at PSI are listed in table 8.1.

Setup	**Events**	**Fluence** p/cm^2	**B13 circuits**	**Cross section** $cm^2/B13$
No TMR	47	2.53 x 10^{10}	100	1.86 x 10^{-11}
TMR	17	1.44 x 10^{11}	33	3.59 x 10^{-12}

Table 8.1: CRAM result for the GateMate A1

A comparison of the results with two other devices is shown in figure 8.2. The Stratix5 and Kintex7 are devices from a comparable technology node [15] and the last three devices in the table are flash memory FPGAs previously tested at CERN [26]. The NG Medium is also tested at CERN [25], but is a fully radiation hardened SRAM based FPGA.

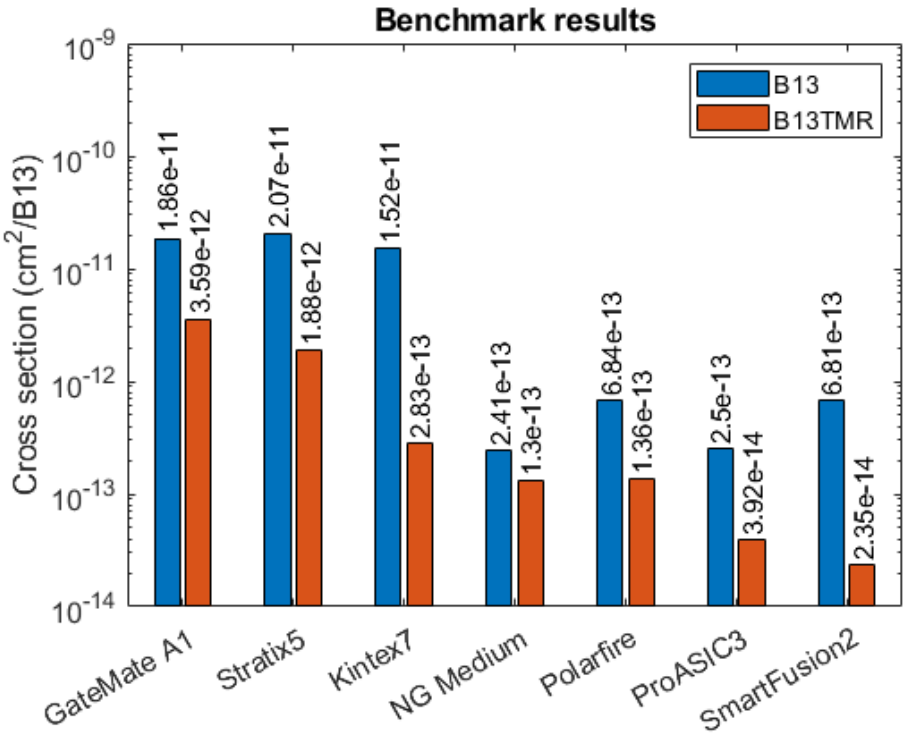


Figure 8.2: B13 results compared to other tested FPGAs.

The GateMate results are comparable to other devices of the same technology node. However, the Stratix5 and Kintex7 achieve better TMR results due to their TMR approach. It is worth noting that the software of these FPGAs supports the integration of TMR. However, the GateMate does not support this feature in the software, so triplication was performed at block level, which is less effective.

8.1.2 CRAM results

By manually positioning the CPEs, to minimize routing, approximately 68% of them were utilized, as shown in figure 8.3. This results in a chain of 13866 CPEs. A bitflip was injected at the first XOR and the time until the output changed was measured with approximately 12.6us.

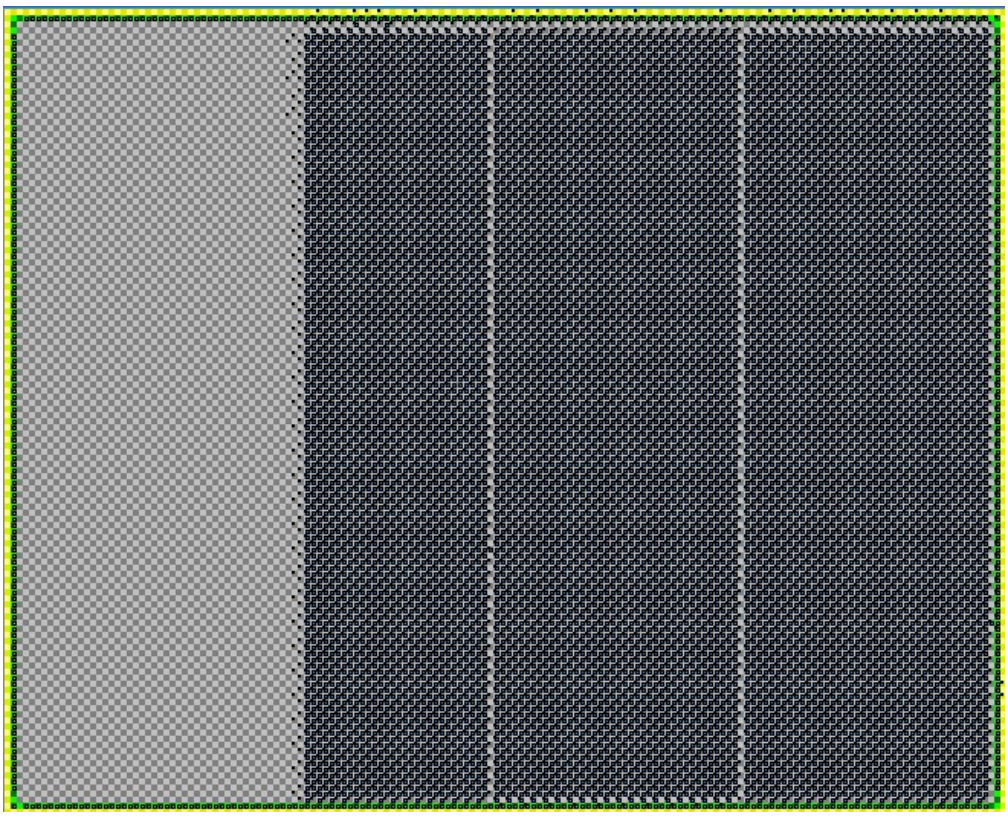


Figure 8.3: Implementation of the CRAM design.

The result of the test under radiation is shown in table 8.2.

Events	**Fluence** p/cm^2	**critical bits**	**cross section** cm^2/Bit
39	2.71 x 10^{10}	199896	7.19 x 10^{-15}

Table 8.2: CRAM result for the GateMate A1

The GateMate test result is compared to two FPGAs from a comparable node type, which are the Stratix5 and the Kintex7 [15] in figure 8.4.

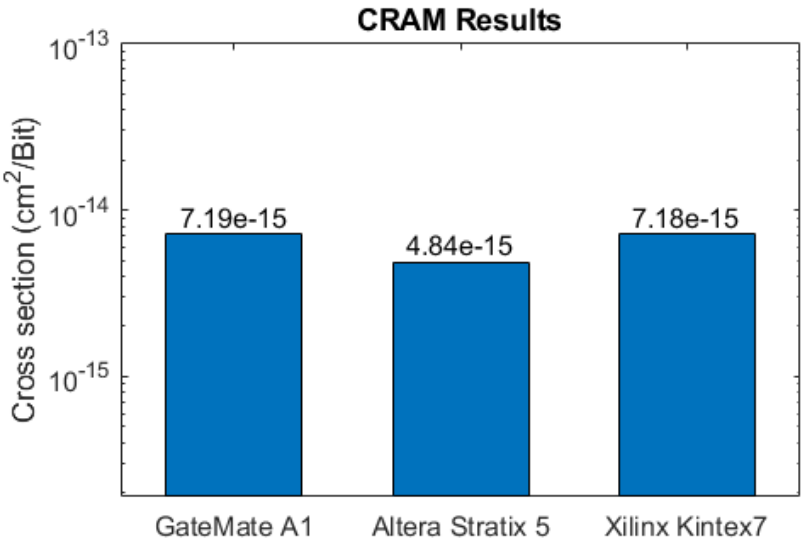


Figure 8.4: Result of CRAM compared to Kintex7 and Stratix5.

These devices show a comparable cross section as already the previous tests showed, which leads to the assumption that the test setup works as assumed.

8.1.3 TID results with focus on relative frequency shifts

The frequency of each ring oscillator is compared by subtracting its frequency from each of the other oscillators. This process is repeated for all dose steps, and the resulting relationship is checked to determine whether it remains positive or negative over the full range. This procedure gives rise to a total of 8385 comparisons between the ring oscillators at each dose step. This number can be calculated using equation 8.1, where N represents the number of implemented ring oscillators.

$$N * (N + 1)/2 - N \tag{8.1}$$

Due to the large amount of data, the dose steps are set to 50Gy and the range is divided into three steps of 3kGy, from 0Gy to 9kGy. Furthermore, the evaluation is divided into two comparisons: the first compares each frequency relation at a dose level to the relation at 0Gy, while the second compares each subsequent dose step to the previous one. This approach allows for an analysis of how many times the relation changes compared to the starting frequency and how many times it changes from one dose level to the next. The resulting graphs are listed in a larger version in the appendix due their size. Figure 8.5 displays the comparison results from 0Gy to 3kGy, with the changes compared to the 0Gy starting point. The percentage of changes is given in relation to the

total number of 8385 comparisons at each dose level. The blue bar shows the percentage of changings to a positive relation and the orange bar a change to a negative relation.

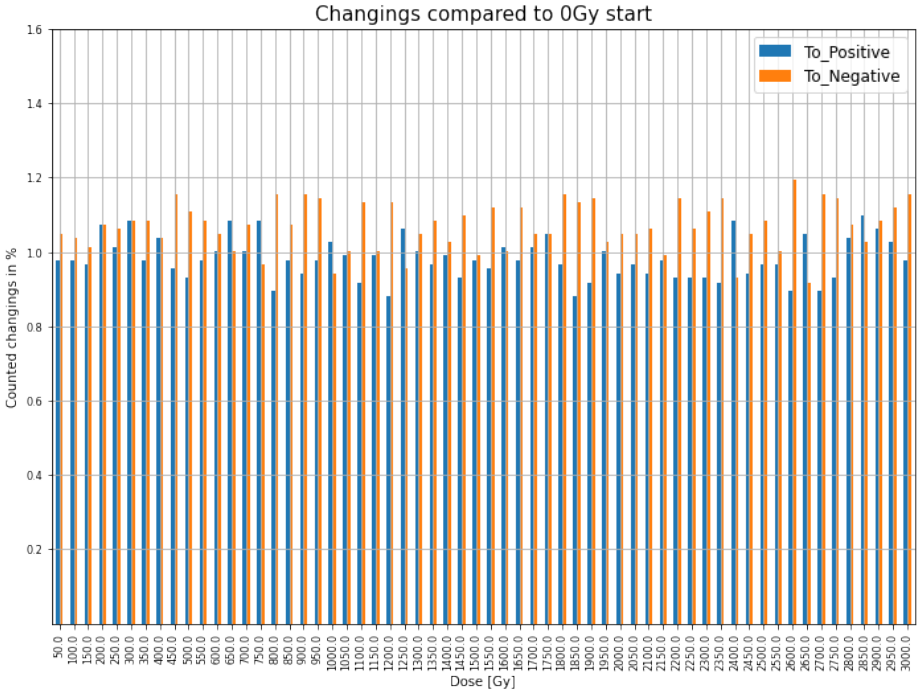


Figure 8.5: Amount of changing relations from 0Gy to 3kGy compared to the 0Gy configuration.

At the outset, there are nearly equal amounts of positive and negative relations, with 4215 being positive and the remainder being negative. The data shows that the total amount of change varies by approximately 1% for both negative and positive relations. This can be seen for the full range of up to 9kGy. There is only a slight increase of 1.1% with rising dose level. The two graphs for 3kGy to 6kGy and 6kGy to 9kGy are shown in appendix B.

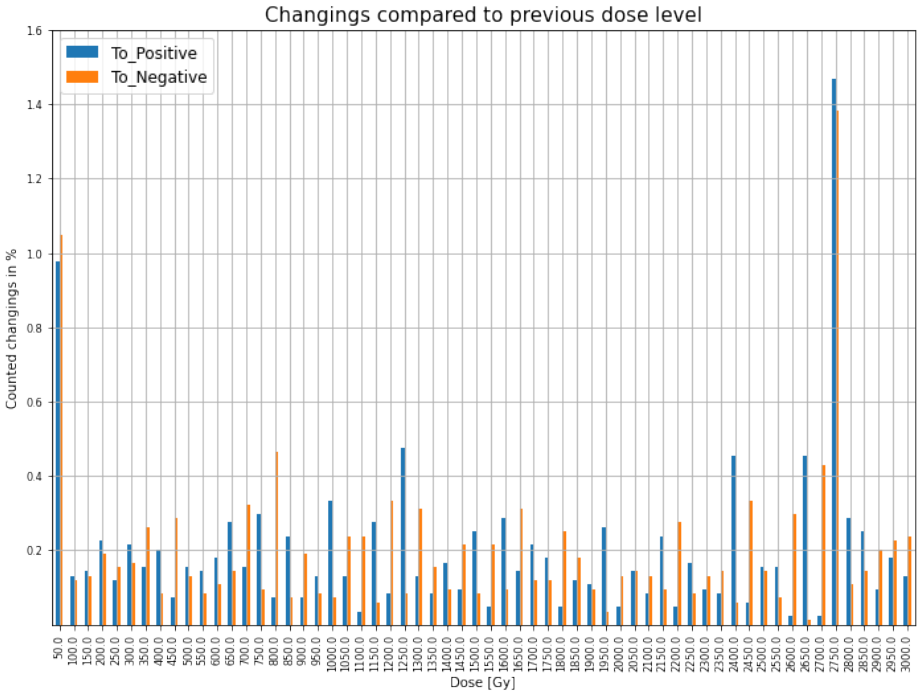


Figure 8.6: Amount of changing relations from 0Gy to 3kGy compared to each previous configuration.

When comparing the change from one dose level to the next, a lower percentage is observed, as shown in figure 8.6. About 0.15% up to 0.4% of the comparisons are changing either to negative or positive, which is a relatively low value. However, it is important to note that the oscillators that change are mostly different. Attempts to filter the ring oscillators for those with a higher impact on the number of changes did not result in a significant reduction. The number of oscillators that stay consistent is lower than the ones that change. Therefore a graph is used to display which ring oscillators exhibit a consistent frequency relation throughout the entire measured dose. A 130x130 comparison matrix is used. A minus (-) indicates a consistent negative relation over the measured dose, a plus (+) indicates a positive relation and an x indicates a changing relation. The matrix is shown in figure 8.7 and is furthermore added to the appendix B. The matrix shows that the inconsistent comparisons are scattered throughout the matrix. In the end only, 16 oscillators demonstrate a consistent relation with all others across the full dose range.

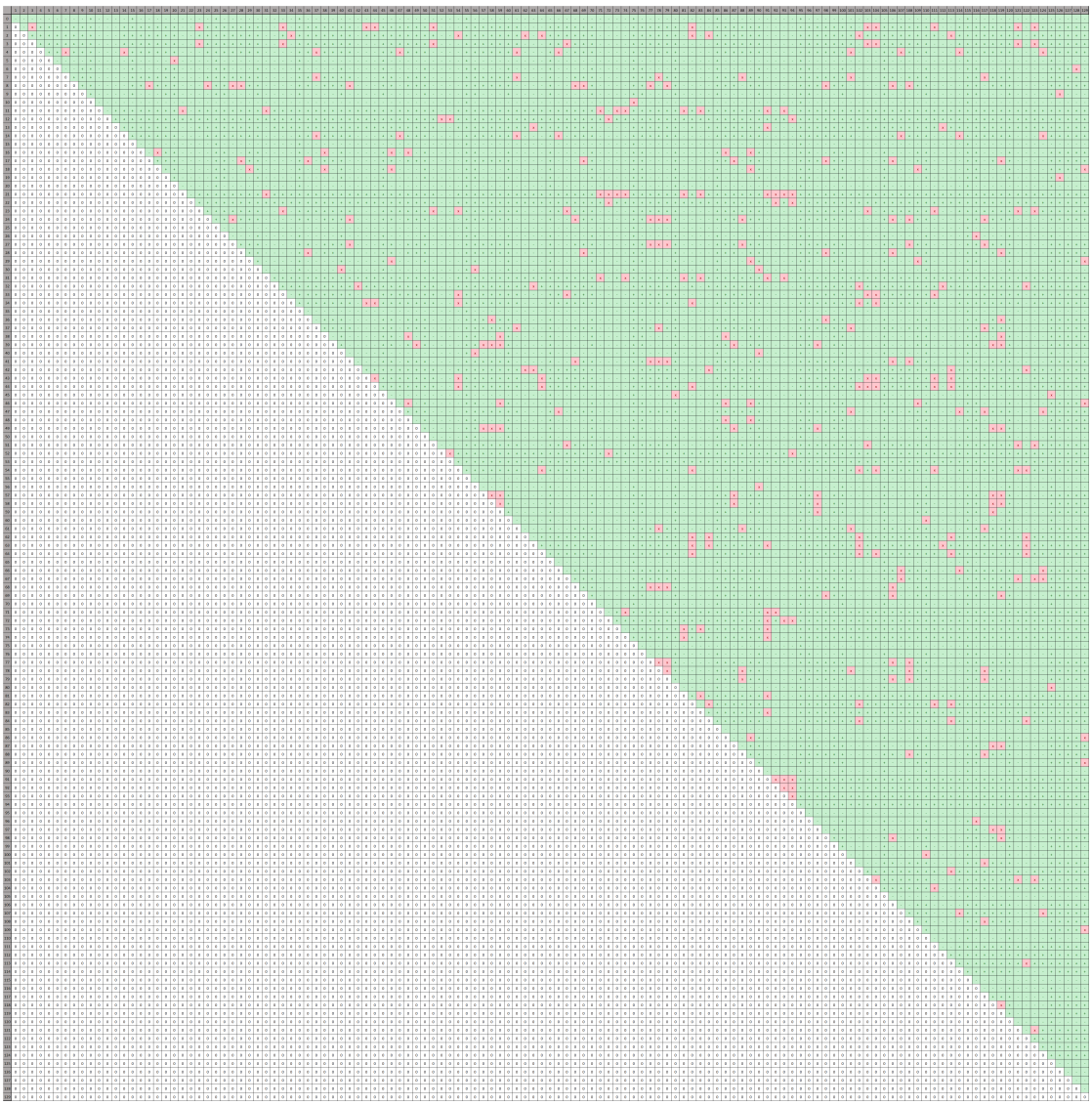


Figure 8.7: Comparison matrix of all ring oscillators from 0Gy to 9kGy.

For further analysis, this matrix is also set up with 3kGy intervals as previously done. The results are attached in the appendix B. Over the dose range of 0Gy to 3kGy, 18 ring oscillators remained consistent, which is only two more than over the full dose range. For the range from 3kGy to 6kGy, 40 remained consistent, and for the last range from 6kGy to 9kGy, 32 remained consistent. Therefore, the highest impact and changes occur in the dose level up to 3kGy and after this, the effect seems to decrease.

Figure 8.8 displays five ring oscillator frequencies. This example shows one oscillator that has a consistent relation to all others and two that are inconsistent. Only the ring oscillator 76 in the middle of the graph shows a consistent relation due to its higher frequency difference related to the others. It should be noted that this is just an excerpt of the complete data. The other 125 ring oscillators show frequencies in the range from 6.2kHz to 7.4kHz at the 0Gy start. In order to enhance readability, the number of displayed frequency graphs has been significantly reduced. A closer look to the relation of two ring oscillators is shown in figure 8.9. It can be seen, that the frequency of these two oscillator are crossing the line of each other multiple times. For a perfect results they should never cross, which is not the case.

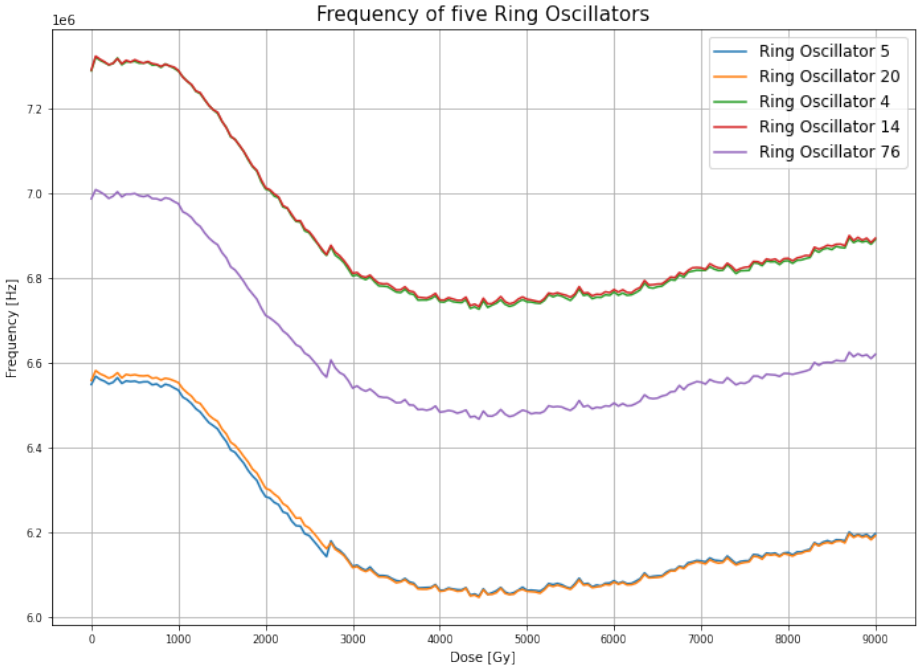


Figure 8.8: Frequency of five ring oscillators over the dose.

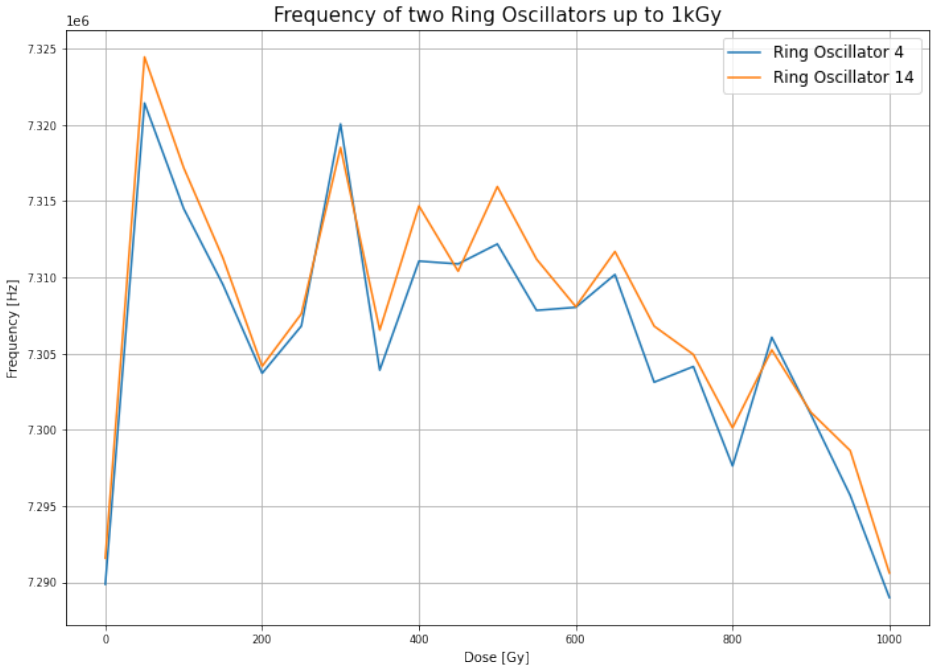


Figure 8.9: Frequency of two ring oscillators from 0Gy to 1kGy.

A further comparison matrix is set up at a dose level of 2.5kGy, as shown in the appendix B, due to the observed peak in frequency at a dose level of approximately 2.7kGy (figure 8.8). This peak could also be observed in the current consumption of the test setup. Up to 2.5kGy, 30 ring oscillator exhibit a consistent relation to the others, which is similar to the range of relations observed from 3kGy up to 9kGy. The reason for this peak in the results in unknown and should be considered for a possible repetition of the test.

The percentage of changing ring oscillators compared to the total number of comparisons is approximately 2%, which is quite low. However, for a perfect result, no oscillator should alter the frequency relation to another one. Filtering the data for chains that have a strong impact on the variation did not prove effective. Although some chains have a higher impact than others, it is not sufficient to consider them as a runaway or faulty chain that should be excluded from the calculation. In the end, the number of changing relations between the ring oscillators shows a consistent number throughout the entire radiation dose.

8.2 iCE40 UltraLite

The iCE40 UltraLite FPGA was tested in one radiation campaign in November 2023. The Breakout Board with the iCE40 UltraLite can be seen in figure 8.10 after integration in the test setup at PSI.

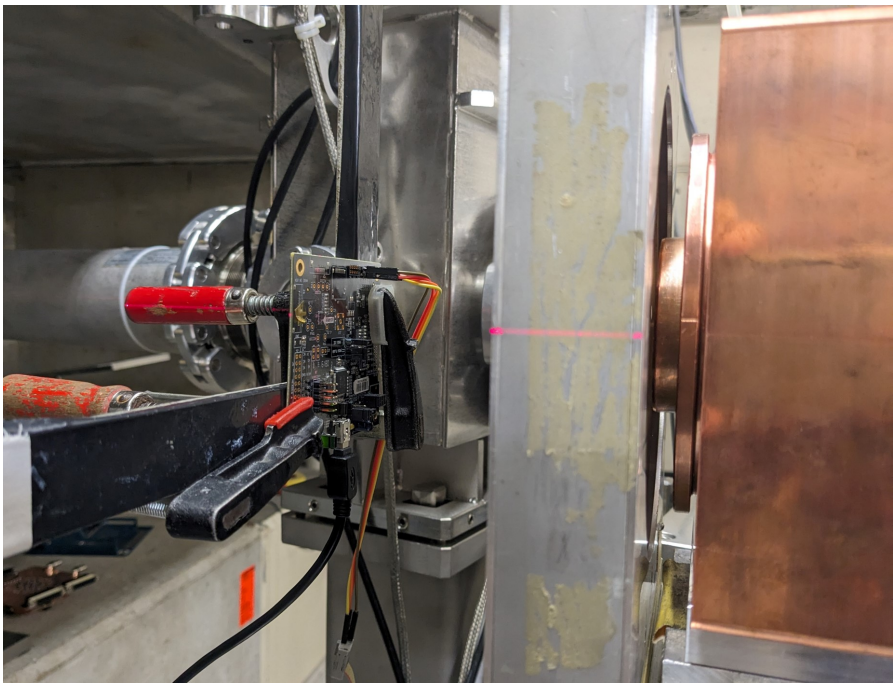


Figure 8.10: Picture of the iCE40 UltraLite at PSI.

The test was done with three setups. A standard version without triplication, a version with TMR and a version where the FPGA was filled as much as possible with watchdog instances. Therefore a total of 12 watchdogs where implemented and voted to one output to check if the behaviour becomes more robust by this approach. This version is named as „Full FPGA“. No events were observed for the standard version of the watchdog test. This is why only the upper limit of the cross section could be calculated. The results are displayed in table 8.3.
Figure 8.11 shows that the TMR version performs slightly better than the version where the chip was filled as much as possible. The attempt to vote the signal from more outputs, not have led to an improvement of the results. Additionally, this version almost completely filled the space of the chip, increasing the likelihood of hitting a critical element compared to a smaller, more compact design.

Setup	**Events**	**Fluence** p/cm^2	**Lower Limit** cm^2/Bit	**cross section** cm^2/Bit	**Upper Limit** cm^2/Bit
No TMR	NaN	1.44 x 10^{11}	NaN	NaN	2.56 x 10^{-11}
TMR	1	2.20 x 10^{11}	9.17 x 10^{-14}	4.54 x 10^{-12}	2.53 x 10^{-11}
Full FPGA	4	6.38 x 10^{10}	1.67 x 10^{-12}	6.27 x 10^{-12}	1.61 x 10^{-11}

Table 8.3: Watchdog results for the iCE40 UltraLite.

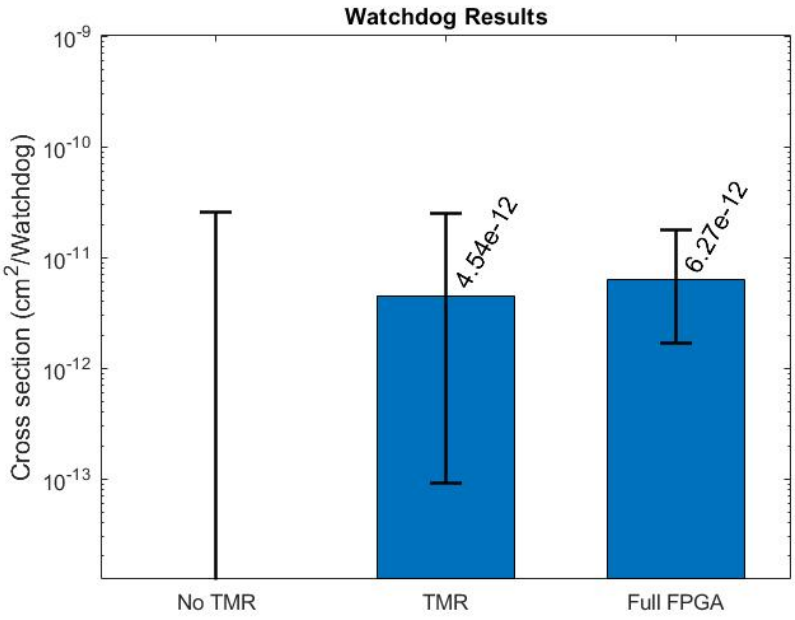


Figure 8.11: Results of the Watchdog test.

The results themself show a low cross section for this application test, but with a higher uncertainty due to the low number of observed events. These test results can serve as a basis for additional investigations on the FPGA.

9 Conclusion and Outlook

In this master thesis, a general overview of radiation effects on electronics together with common mitigation techniques has been given. Following this, a brief introduction to FPGAs is provided, along with a detailed description of the GateMate FPGA and the iCE40 UltraLite. The devices are tested in different setups. First, an environment for measuring the leakage current of the GateMate FPGA was set up and corresponding leakage current measurements were carried out. The radiation qualification tests at CERN where explained, finished and evaluated. The iCE40 UltraLite is tested in a first application test, which is a basic setup of the function for which this FPGA could be used for. The results are also evaluated.

The leakage current of the GateMate FPGA is measured for two different types of produced chips. One is a short-loop production and one is a standard type. The results of this test show a clear difference between the types. The short-loop device has a higher leakage current compared to the second FPGA-chip. This can be explained by the manufacturing process. As the name of this process suggests, the short-loop chip does not go trough the whole production process like a standard full-loop chip. The difference is clearly visible in the results shown in figure 6.4 and 6.5. As expected, the curve for both devices shows an exponential increase in leakage current as a function of temperature without any outliers.

The test designs of the GateMate were focused on the benchmark test B13 and the CRAM configuration test, which are the outstanding radiation tests for the GateMate FPGA. The benchmark B13 test was prepared in combination with a Zedboard and the setup was transferred to the MicroZedboard at CERN. The design included 100 B13 instances for the non TMR version and 33 B13 instances for the TMR version. The sensitivity of the CRAM was tested using a design of an XOR chain. Both tests resulted in a low cross section, comparable to other FPGAs of this technology node. The radiation qualification of the GateMate at CERN has been completed and a brief summary of the results has been documented in a paper. Additionally, this paper was submitted to the Nuclear and Space Radiation Effects Conference (NSREC) in Canada in July of 2024 and was accepted.

The results of the previously conducted TID test were re-evaluated, with focus on the frequency relation between all ring oscillators across the measured dose range for the use

in a Physically Unclonable Function (PUF). The data indicate a consistent number of relations that change throughout the entire dose range. Ideally, no relation should change. In fact only about 2% of the total comparisons show changes. Additionally, some of the oscillator frequencies are very close to others. In these cases, even a small frequency change can lead to a changed comparison result. An interesting continuation of the test for use in PUFs, would be the implementation of ring oscillators and the observation of their behavior at different temperatures. The most stable oscillators could then be used for a comparison under irradiation, to determine if their behavior remains consistent.

The results obtained from the Lattice iCE40 UltraLite FPGA provide an intriguing starting point for further investigation at CERN. The cross section of the watchdog setup is in a low range giving motivation for further tests. Therefore, it is necessary to implement the test setups as they were used for the GateMate irradiation qualification also for the iCE40 UltraLite to evaluate the performance of this FPGA.

List of Figures

Bibliography

[1] G. Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 1–29. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.020. URL: https://www.sciencedirect.com/science/article/pii/S037026931200857X.

[2] *About the LXI Consortium - Sponsored and Developed by Leading Test & Measurement Companies*. 14.11.2023. URL: https://www.lxistandard.org/about/AboutLXI.aspx.

[3] Hideharu Amano, ed. *Principles and Structures of FPGAs*. 1st edition 2018. Puchong, Selangor D.E.: Springer Singapore and Springer, 2018. ISBN: 9789811308239.

[4] Melanie Berg. "Field Programmable Gate Array (FPGA) Single Event Effect (SEE) Radiation Testing". In: (). URL: https://nepp.nasa.gov/files/23779/fpga_radiation_test_guidelines_2012.pdf.

[5] CERN. *ALICE*. 5.11.2023. URL: https://home.cern/science/experiments/alice.

[6] CERN. *The Large Hadron Collider*. 5.11.2023. URL: https://home.cern/science/accelerators/large-hadron-collider.

[7] Marco Calviani. *Raditation Hardness Assurance: Basic introduction to radiation effects on electornics*. URL: https://indico.cern.ch/event/1287100/contributions/5408902/attachments/2661918/4611873/RHA.pdf.

[8] Cologne Chip AG. *GateMate (TM) FPGA Datasheet*. URL: https://colognechip.com/docs/ds1001-gatemate1-datasheet-latest.pdf.

[9] Cologne Chip AG. "GateMate(TM) FPGA Evaluation Board Datasheet". In: (). URL: https://colognechip.com/docs/ds1003-gatemate1-evalboard-3v1-latest.pdf.

[10] F. Corno, M. S. Reorda, and G. Squillero. "RT-level ITC'99 benchmarks and first ATPG results". In: *IEEE Design & Test of Computers* 17.3 (2000), pp. 44–53. DOI: 10.1109/54.867894.

[11] *GateMate FPGA Evaluation Board – Cologne Chip*. URL: https://colognechip.com/programmable-logic/gatemate-evaluation-board/.

[12] Import. "Welcome to the Proton Irradiation Facility (PIF)". In: *Paul Scherrer Institut (PSI)* (6.05.2014). URL: https://www.psi.ch/en/pif.

[13] Paul Jones. *Navigating Efficiency: Exploring The Advantages Of Short Loop Wafers In Processing - I Stock Analyst*. 2023.

[14] Richard Jung. *Radiation Qualification of the Cologne Chip GateMate A1 FPGA*. 2023. DOI: 10.26205/OPUS-3364. URL: https://opus.bsz-bw.de/fhdo/frontdoor/deliver/index/docId/3364/file/Master_Thesis_Richard_Jung-1.pdf.

[15] Andrew Keller. "Using On-Chip Error Detection to Estimate FPGA Design Sensitivity to Configuration Upsets". In: *Theses and Dissertations* (2017). URL: https://scholarsarchive.byu.edu/etd/6302.

[16] MUTUEL Laurence. *Single Event Effects Mitigation Techniques Report*. URL: https://www.tc.faa.gov/its/worldpac/techrpt/tc15-62.pdf.

[17] *MicroZed | Avnet Boards*. 10.04.2024. URL: https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/microzed/.

[18] *Mikroelektronik: Halbleiterbauelemente und deren Anwendung in elektronischen Schaltungen*. SpringerLink Bücher. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. ISBN: 9783540267300. DOI: 10.1007/b138008.

[19] Mehdi Mneja. "FPGA-Implementierung einer Ringoszillator basierten Physically Unclonable Function: Bachelorthesis". PhD thesis. Fachhochschule Dortmund, 2023. DOI: 10.26205/opus-3327. URL: https://opus.bsz-bw.de/fhdo/frontdoor/index/index/start/1/rows/10/sortfield/score/sortorder/desc/searchtype/simple/query/gatemate/docId/3327.

[20] PYNQ - Python productivity for Zynq. *PYNQ - Python productivity for Zynq*. 25.03.2024. URL: http://www.pynq.io/.

[21] *Project Jupyter*. 14.03.2024. URL: https://jupyter.org/.

[22] Rudy Ferraro. "Development of Test Methods for the Qualification of Electronic Components and Systems Adapted to High-Energy Accelerator Radiation Environments". PhD thesis. Université Montpellier, 24.06.2020. URL: https://hal.science/tel-02879667.

[23] Antonio Scialdone. "FPGA qualification for the LHC radiation environment". PhD thesis. URL: https://webthesis.biblio.polito.it/19117/.

[24] *The accelerator complex | CERN*. 5.11.2023. URL: https://home.cern/science/accelerators/accelerator-complex.

[25] G. Tsiligiannis et al. "Reliability analysis of a 65nm Rad-Hard SRAM-Based FPGA for CERN applications". In: *2019 19th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*. 2019, pp. 1–8. DOI: 10.1109/RADECS47380.2019.9745713.

[26] Georgios Tsiligiannis et al. "Investigation on the sensitivity of a 65nm Flash-based FPGA for CERN applications". In: *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*. 2016, pp. 1–4. DOI: 10.1109/RADECS.2016.8093209.

[27] *Yosys Open SYnthesis Suite :: About*. 5.12.2022. URL: https://yosyshq.net/yosys/.

[28] *ZedBoard | Avnet Boards*. 10.04.2024. URL: https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/zedboard/.

[29] *iCE40 Programming and Configuration (FPGA_ TN-02001)*. 10.04.2024. URL: http://www.latticesemi.com/view_document?document_id=46502.

[30] *iCE40 UltraLite Familiy Data Sheet*. 10.04.2024. URL: https://www.latticesemi.com/view_document?document_id=50945.

A Appendix: Software and Hardware

Vivado:

- Name: Vivado 2022.2 and 2020.1
- Manufacturer: Xilinx
- URL: `https://www.xilinx.com/products/design-tools/vivado.html`

Vitis:

- Name: Vitis 2022.1
- Manufacturer: Xilinx
- URL: `https://www.xilinx.com/products/design-tools/vitis.html`

iCEcube2:

- Name: iceCube2
- Manufacturer: Lattice
- URL: `https://www.latticesemi.com/iCEcube2`

Diamond Programmer:

- Name: Diamond Programmer
- Manufacturer: Lattice
- URL: `https://www.latticesemi.com/LatticeDiamond`

GateMate:

- Name: GateMate evaluation board V3.1.A
- Software: Toolchain Version 05.2023
- Manufacturer: Cologne Chip
- URL: `https://colognechip.com/programmable-logic/gatemate-evaluation-board/`

Zedboard:

- Name: Zedboard 7020
- Manufacturer: Avnet
- URL: https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/zedboard/

MicroZedboard:

- Name: MicroZed 7010/7020
- Manufacturer: Avnet
- URL: https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/microzed/

iCE40 Ultra Lite:

- Name: iCE40 Ultra Lite Breakout Board
- Manufacturer: Lattice
- URL: https://www.latticesemi.com/products/developmentboardsandkits/ice40ultralitebreakoutboard

Thermostream:

- Name: Thermostream
- Manufacturer: inTEST Thermal Solutions
- URL: https://www.intestthermal.com/temptronic/thermostream

HMC Power supply:

- Name: HMC 8043
- Manufacturer: Rode & Schwarz
- URL: https://www.rohde-schwarz.com/de/produkte/messtechnik/dc-netzgeraete/rs-hmc804x-dc-netzgeraeteserie_63493-61542.html

Digital Multimeter (DMM):

- Name: DMM 6500
- Manufacturer: Keithley
- URL: https://www.tek.com/de/products/keithley/digital-multimeter/dmm6500

B Appendix: TID frequency data

This appendix includes the graphics of the TID data evaluation, including the graphics for:

- Frequencies compared to 0Gy:
  - 0-3kGy
  - 3-6kGy
  - 6-9kGy
- Frequencies compared to previous dose level:
  - 0-3kGy
  - 3-6kGy
  - 6-9kGy
- Comparison Matrix:
  - 0-3kGy
  - 3-6kGy
  - 6-9kGy
  - 0-2.5kGy
- Frequency of five ring oscillators up to 9kGy
- Frequency of two ring oscillators up to 1kGy

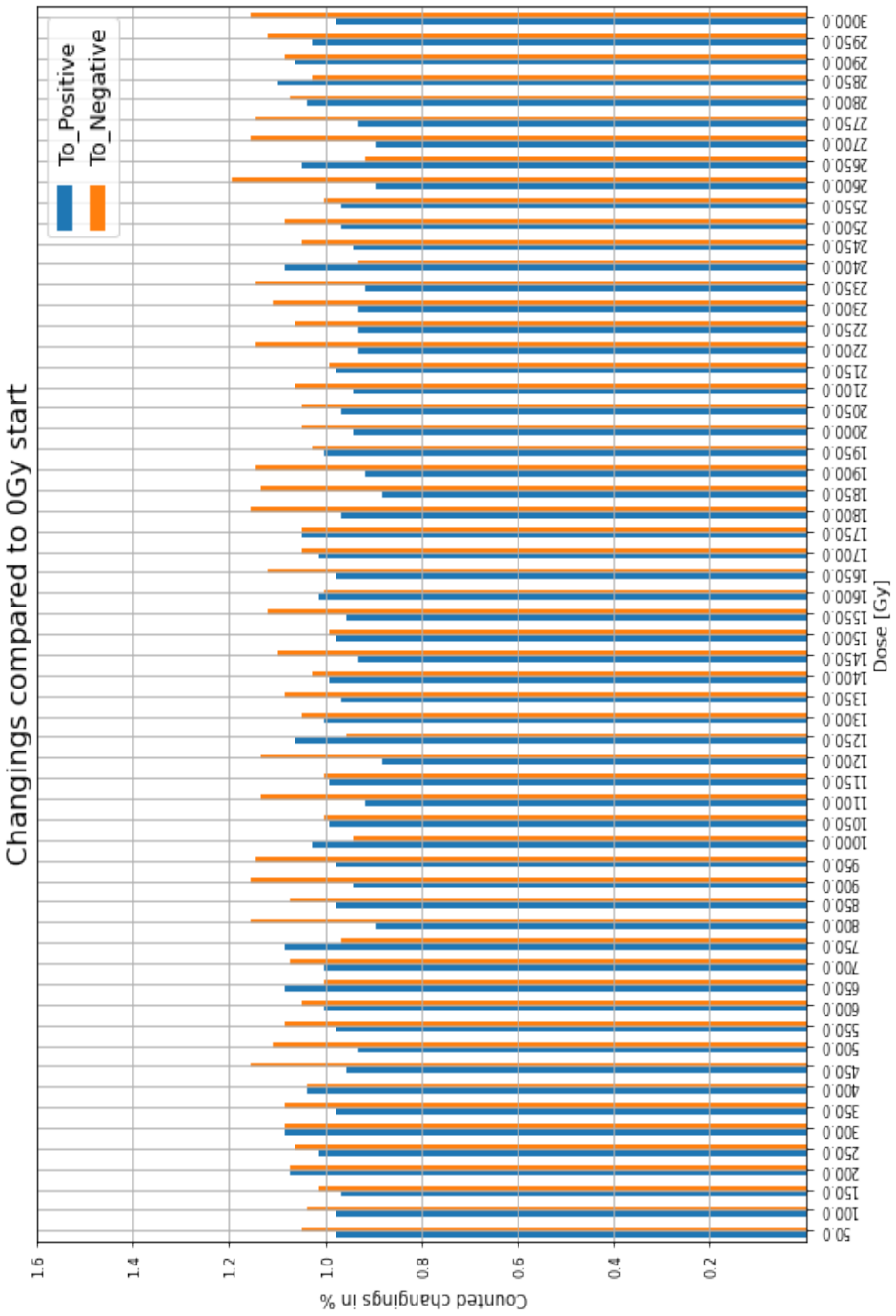


Figure B.1: 0-3kGy compared to 0Gy.

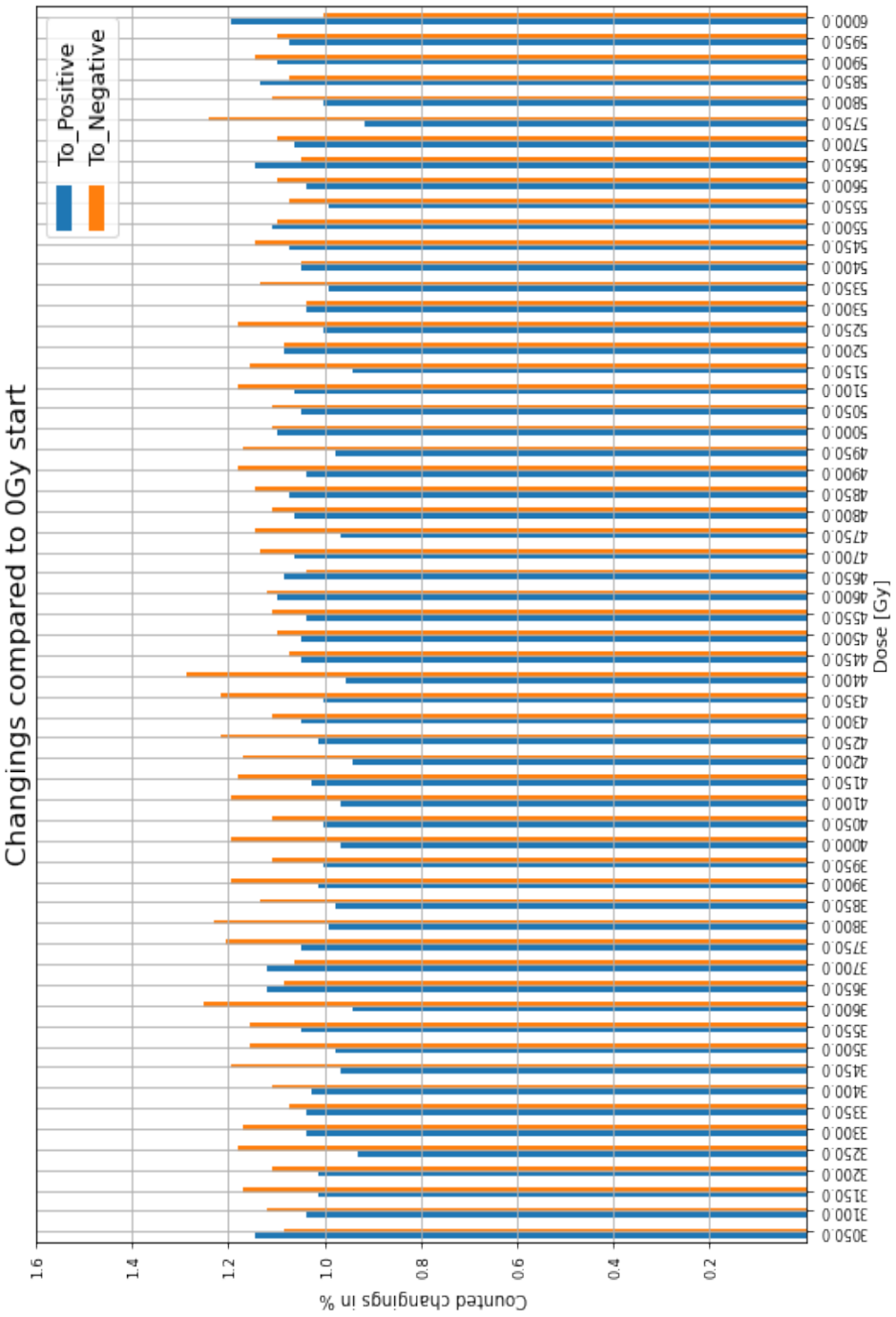


Figure B.2: 3-6kGy compared to 0Gy.

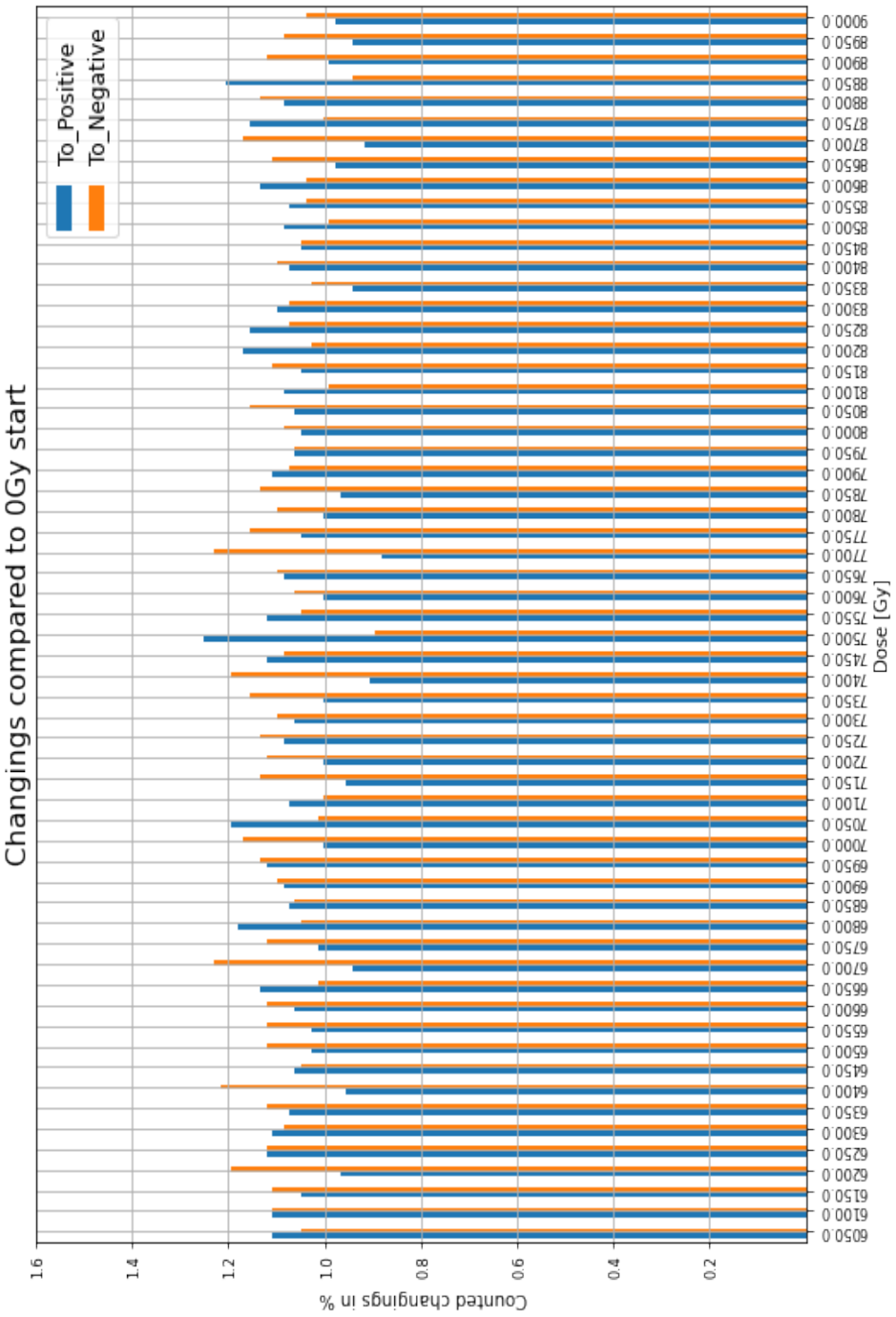


Figure B.3: 6-9kGy compared to 0Gy.

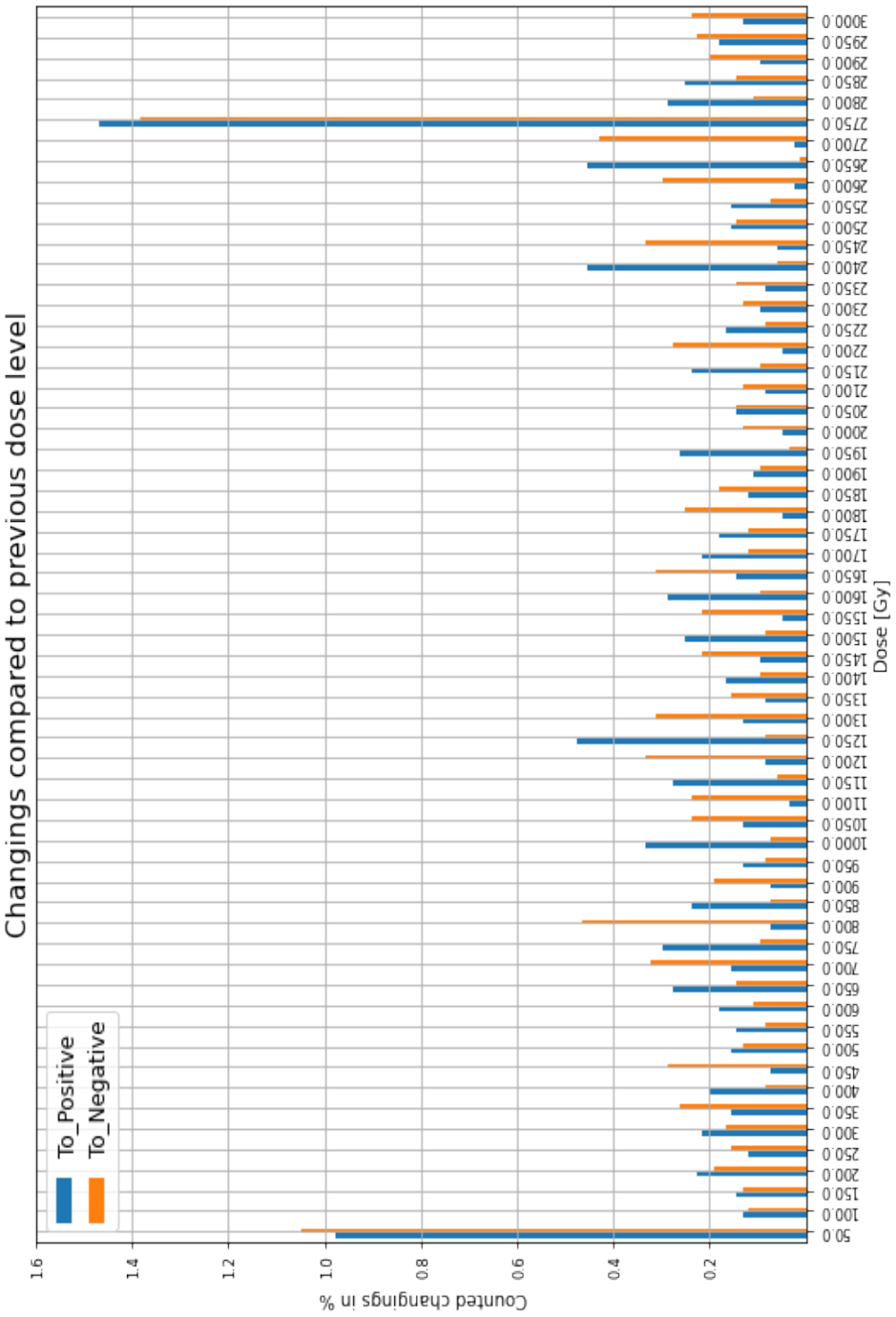


Figure B.4: 0-3kGy compared to previous dose level.

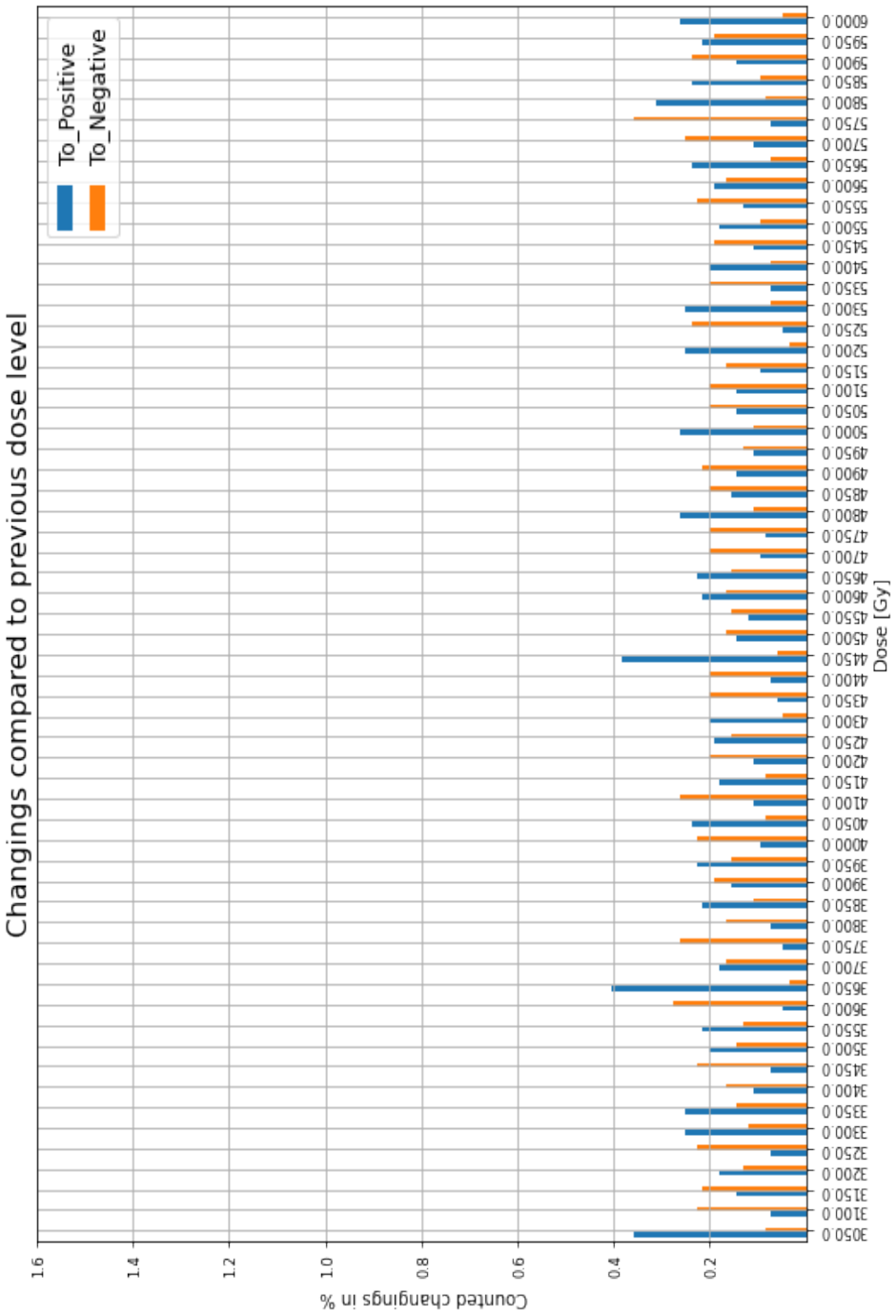


Figure B.5: 3-6kGy compared to previous dose level

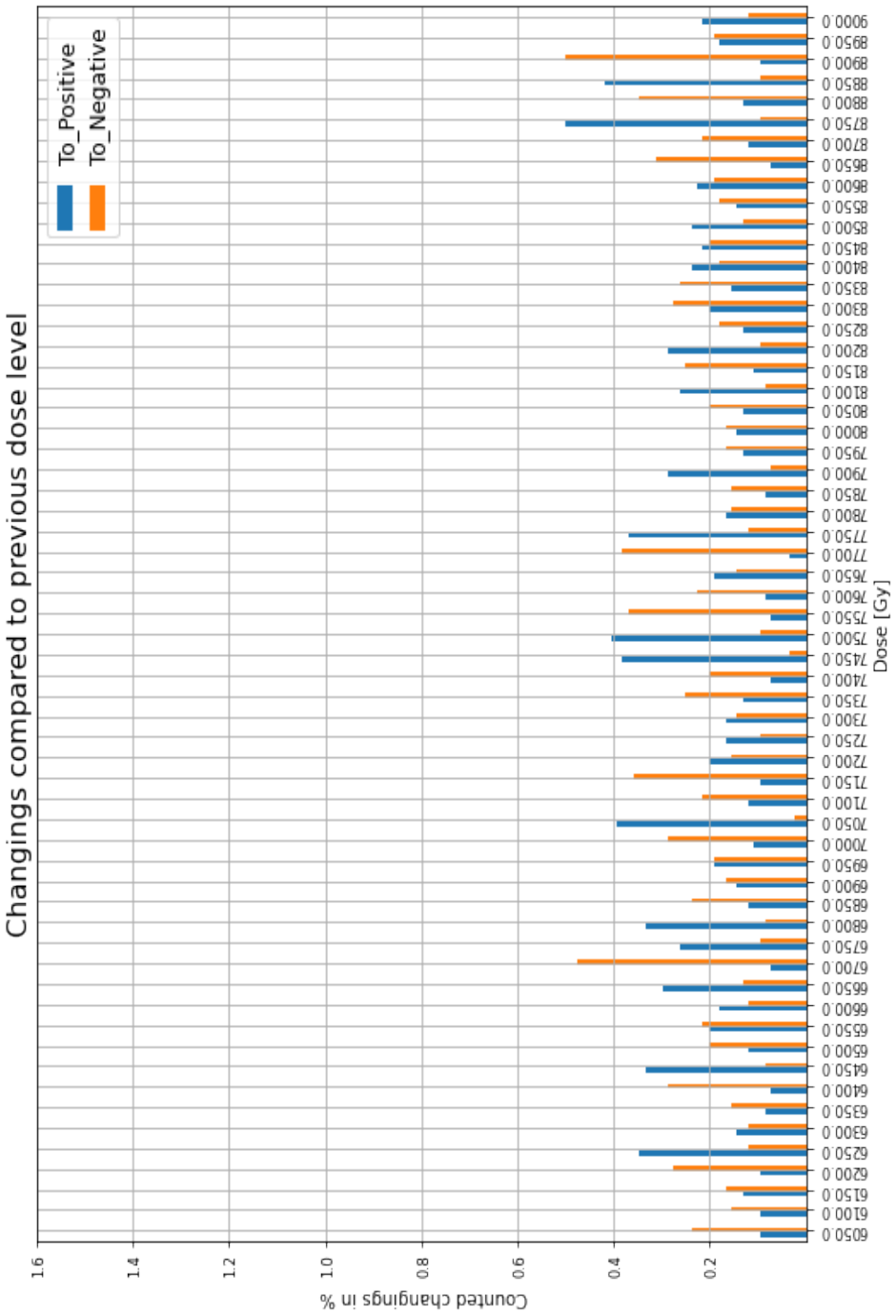


Figure B.6: 6-9kGy compared to previous dose level

Fully compared up to 9kGy

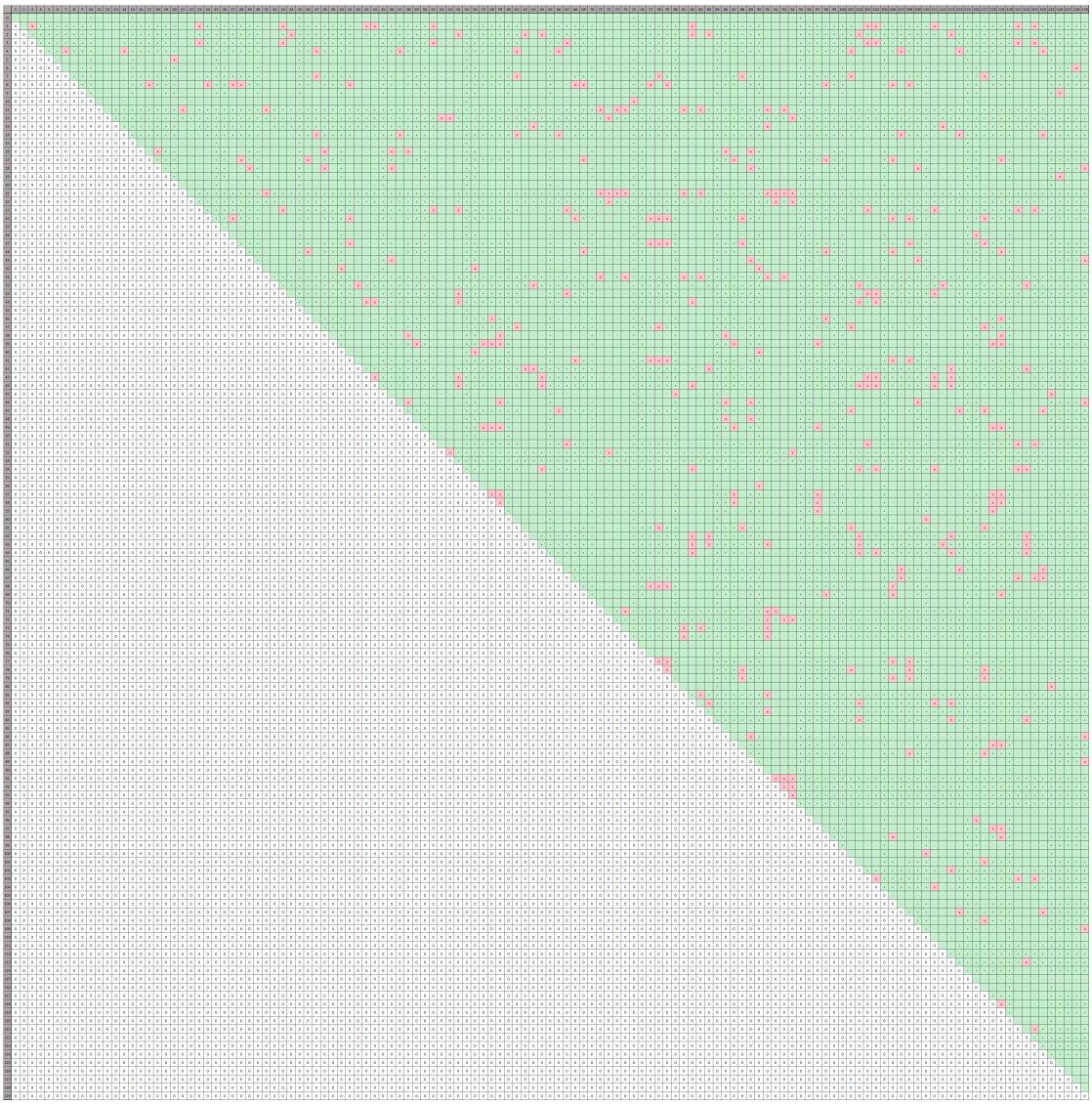


Figure B.7: Comparison matrix of all ring oscillators.

0-3kGy

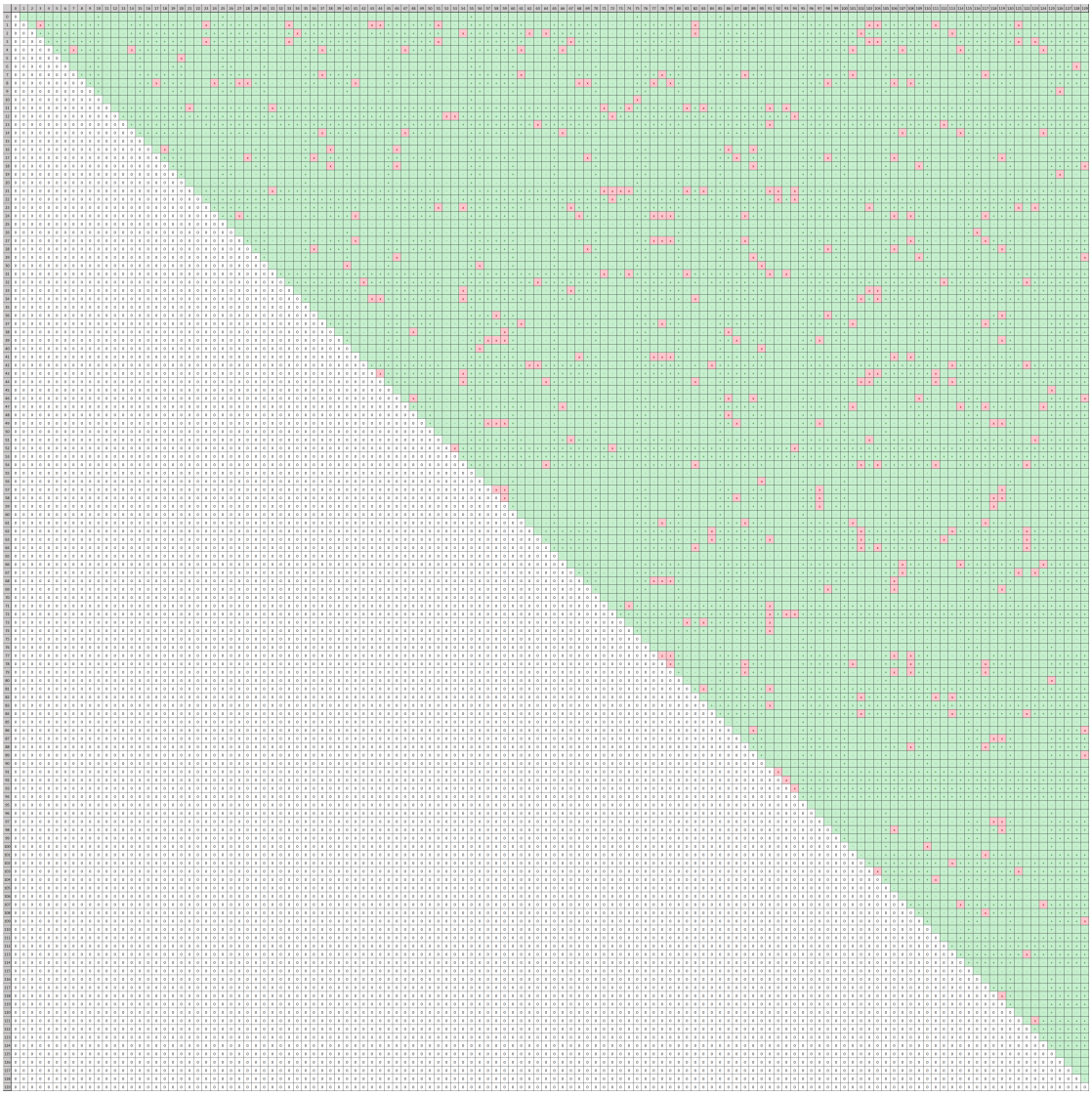


Figure B.8: Comparison matrix up to 3kGy.

3-6kGy

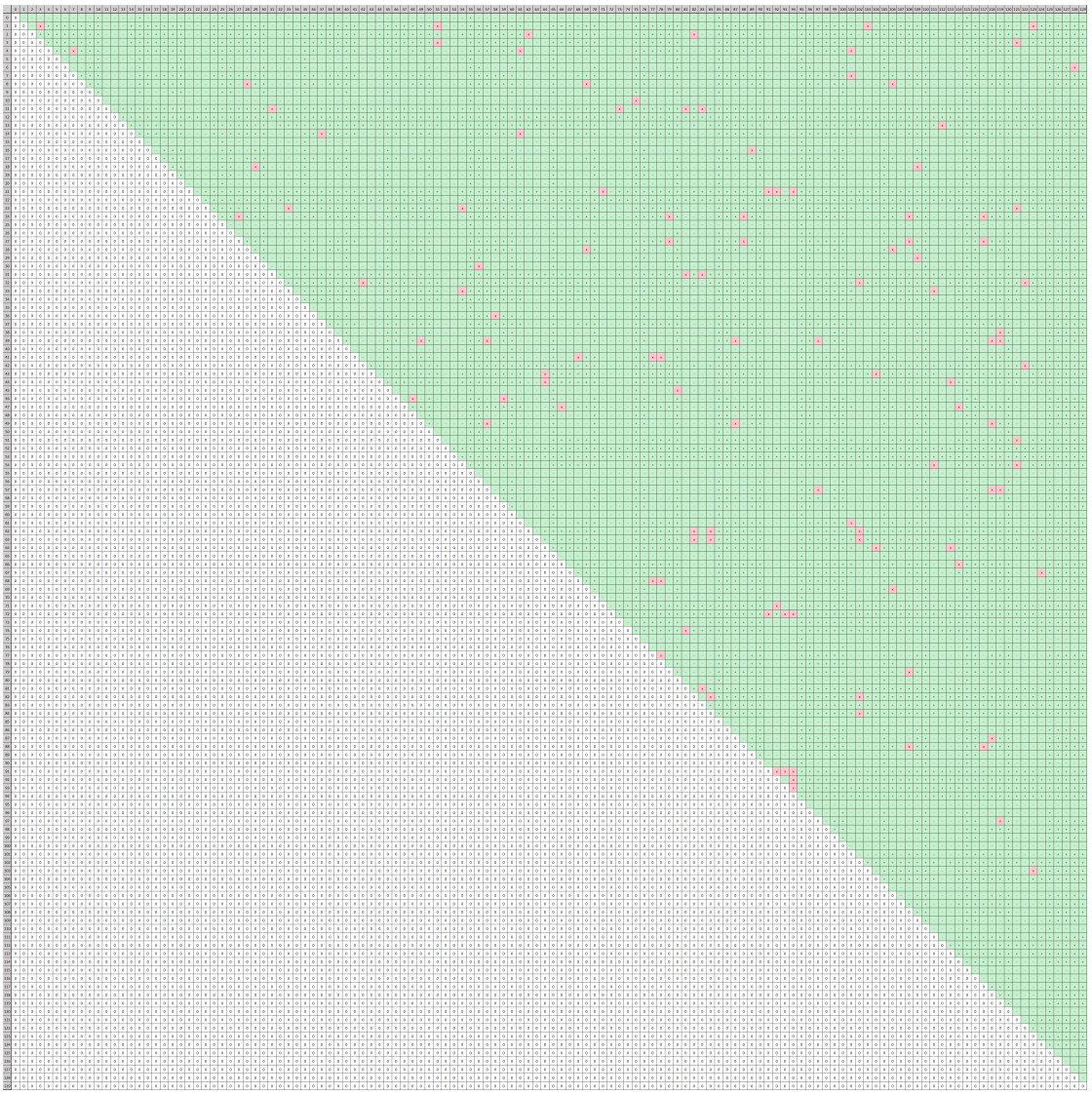


Figure B.9: Comparison matrix from 3kGy to 6kGy.

6-9kGy

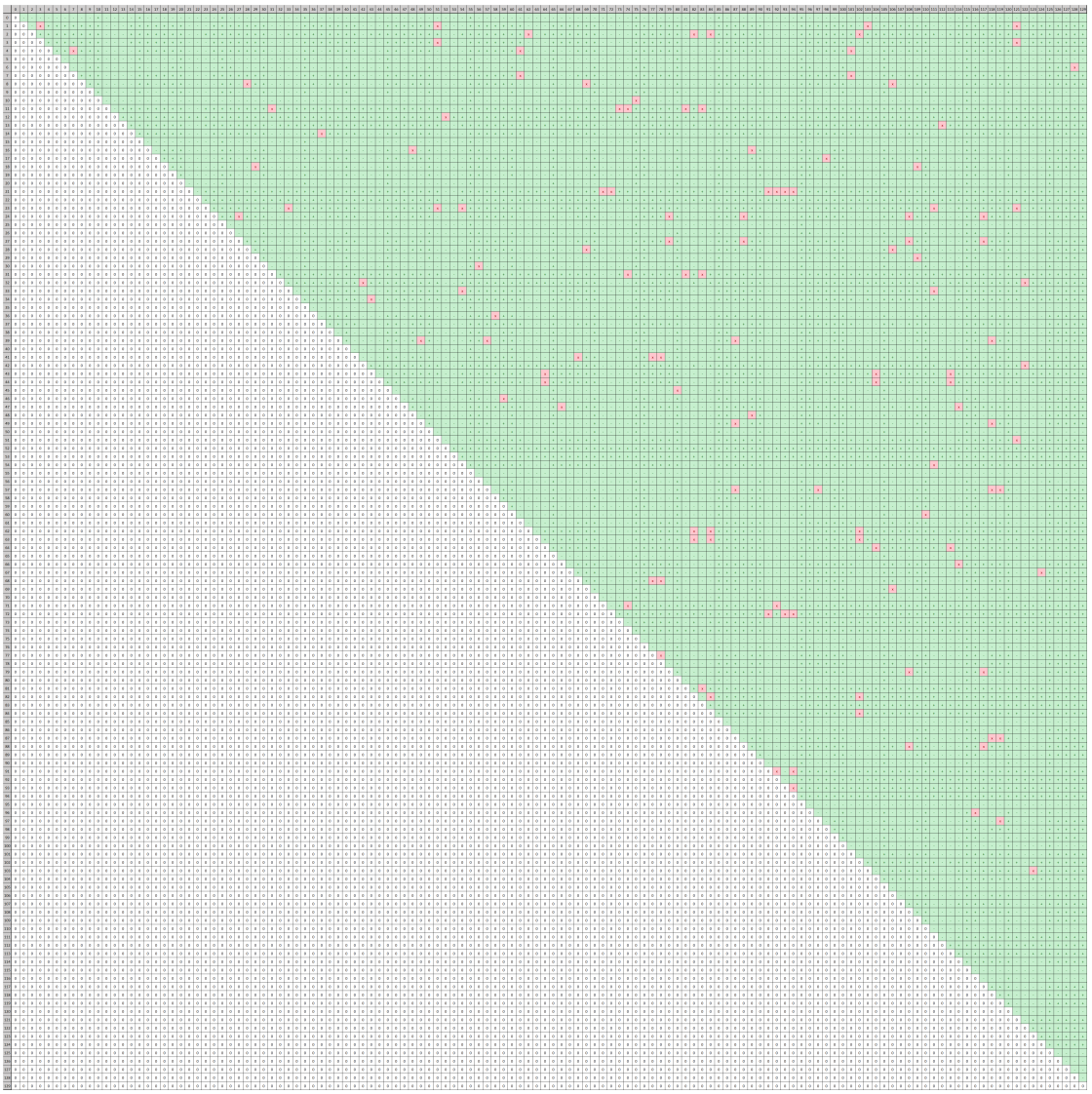


Figure B.10: Comparison matrix from 6kGy to 9kGy.

0-2.5kGy

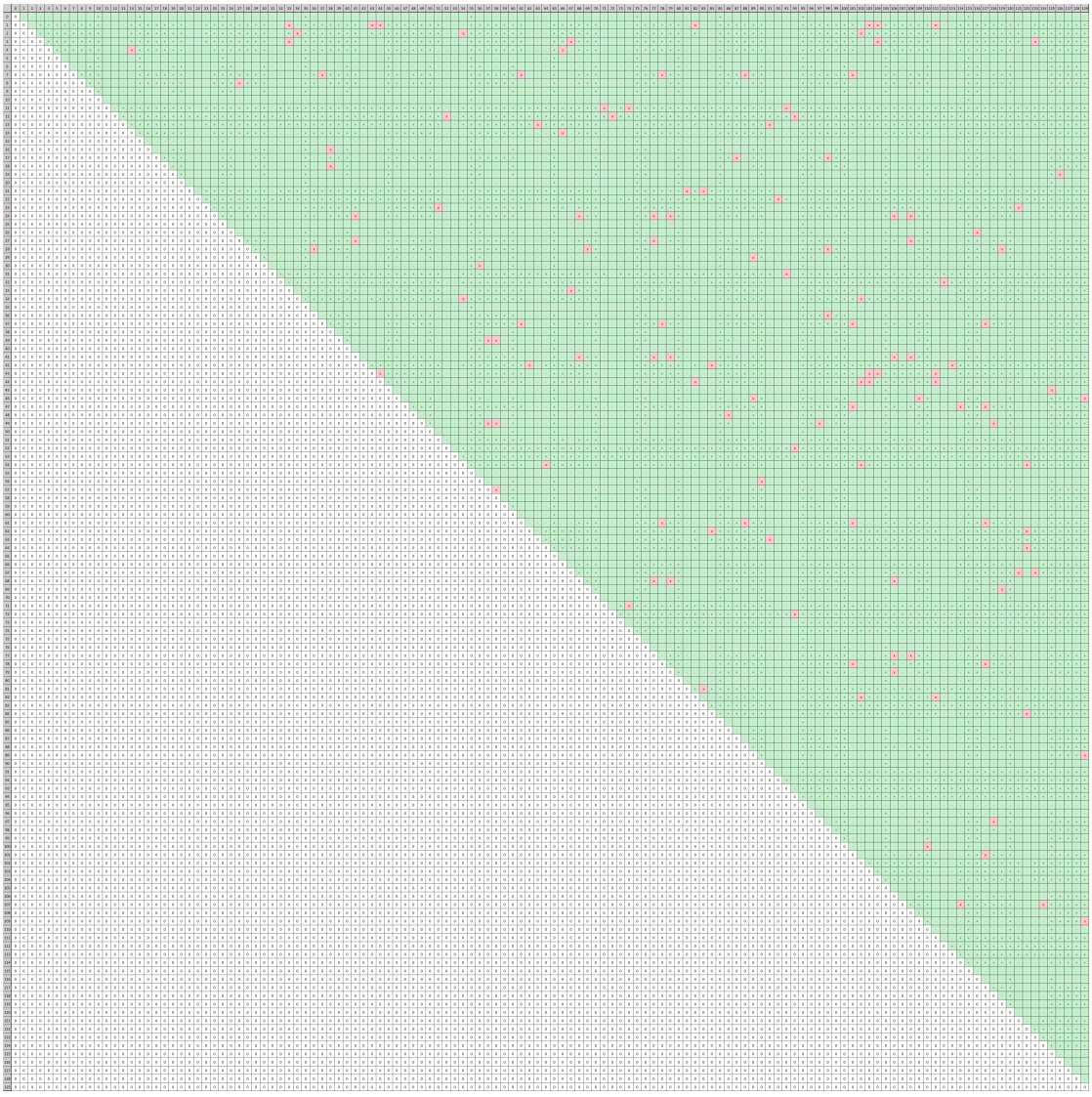


Figure B.11: Comparison matrix up to 2.5kGy.

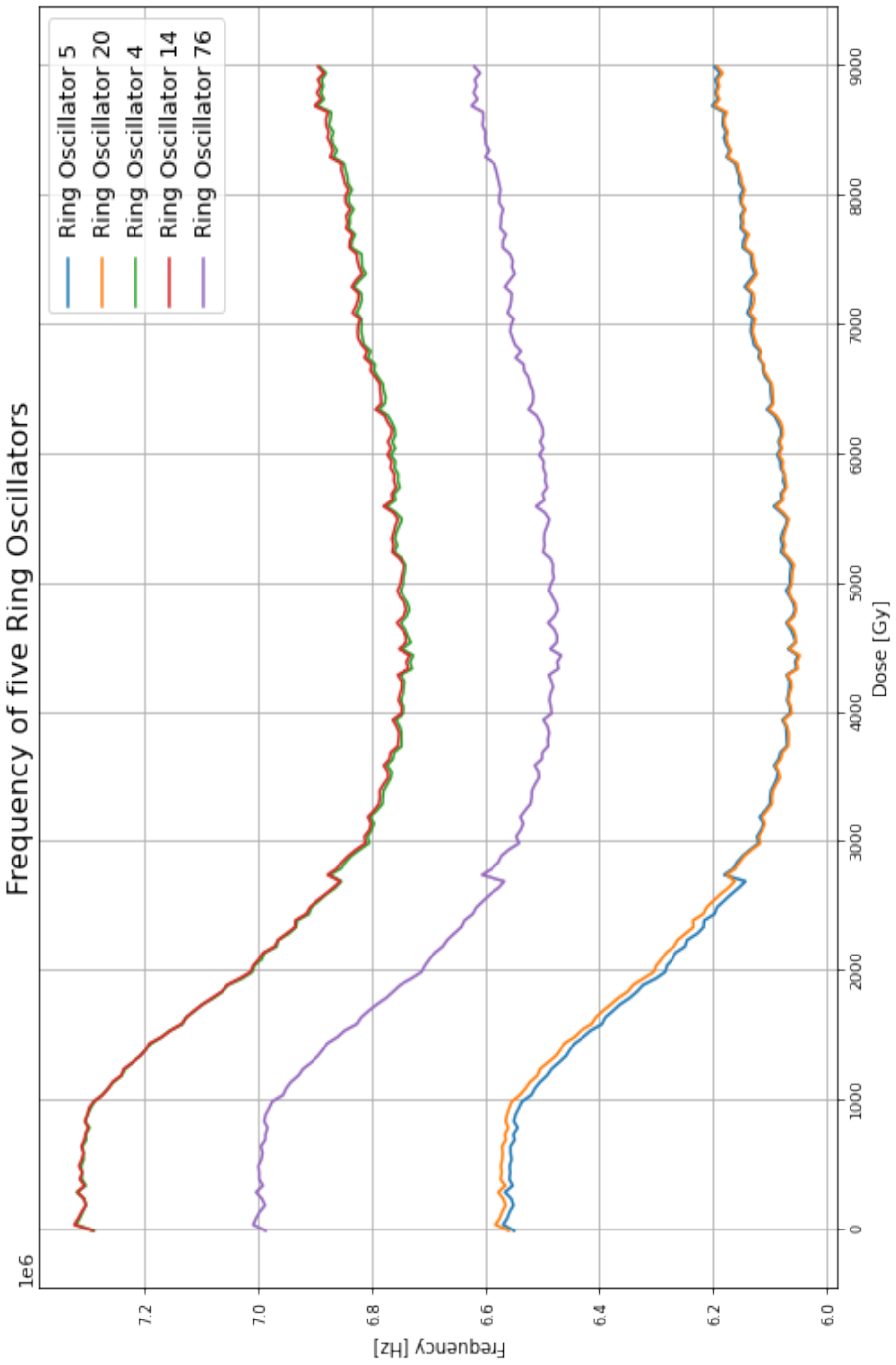


Figure B.12: Frequency of five ring oscillators over the dose.

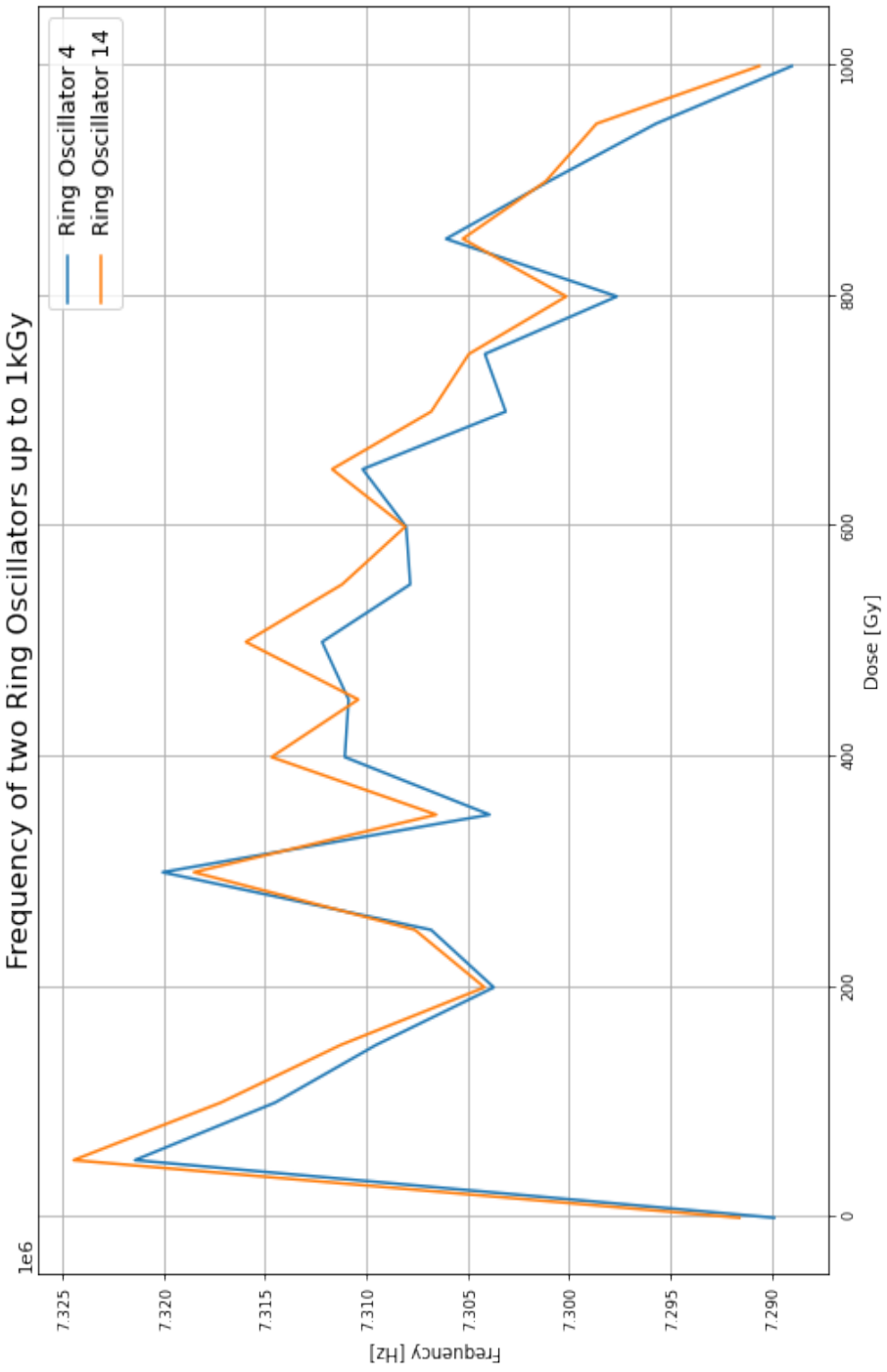


Figure B.13: Frequency of two ring oscillators from 0Gy to 1kGy.

C Appendix: Tutorial

Fachhochschule
Dortmund
University of Applied Sciences and Arts

Guide for integrating IPs into Xilinx Zynq SoCs via Vivado under Peta-Linux and subsequent creation and use of a project platform in Vitis

Part of the Master thesis

Lars Koers

April 2024

Structure of the guide:

The first section discusses how to create an IP as an Axi4-Lite interface in Vivado. A block design is created in Vivado and then exported. At the end of the chapter, an .xsa file is provided for the next installation steps.

In the second chapter, the .xsa file of the hardware design is used to perform the PetaLinux installation. If you already have an .xsa file, you can skip to chapter 2.

In chapter 3, the files created during the installation process are loaded onto a disk which is divided into two partitions. The result is a bootable project on the disk.

The two following chapters 4 and 5 contain test options for the project and a short introduction to commands for adapting the source code.

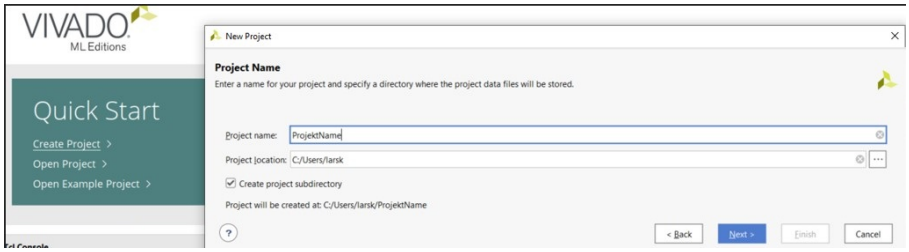
Chapter 6 uses the PetaLinux project to generate a platform in Vitis. Based on this platform an application written in C can be executed.

1 Integration of an IP with a hardware design in Vivado

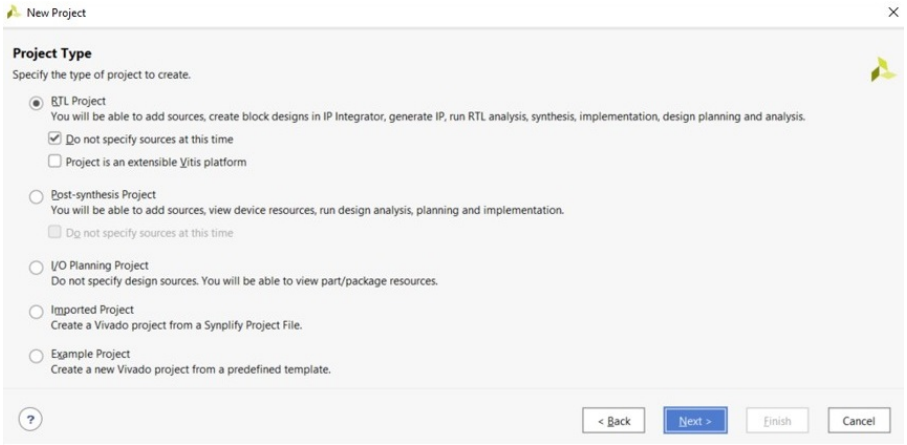
The creation takes place under Vivado 2022 and serves as an example to illustrate the procedure.

1.1 Create and package new IP

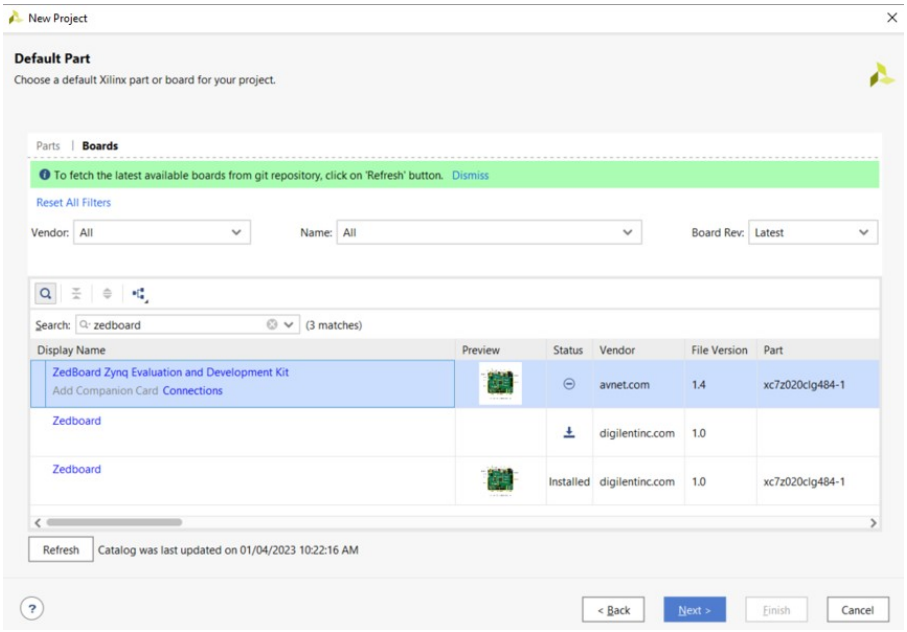
1.1.1 The first step is to create a new project in Vivado and assign a name to it.



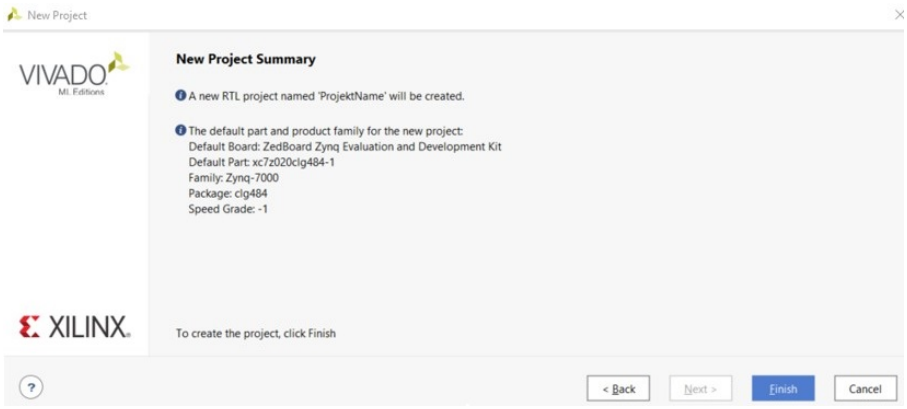
1.1.2 Choose RTL project and go on.



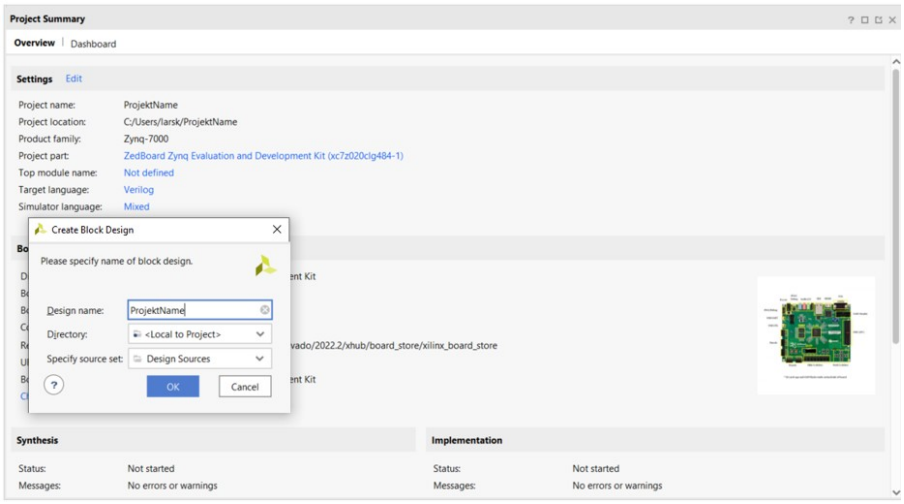
1.1.3 Choose the board for the project.
→ !Attention! please make sure that the correct board file is selected.
→ If necessary, the catalog should be updated. → Refresh



→ In this example, the Zedboard is used: `ZedBoard Zynq Evaluation and Development Kit` from avnet.com
1.1.4 The project can be created with the following settings.



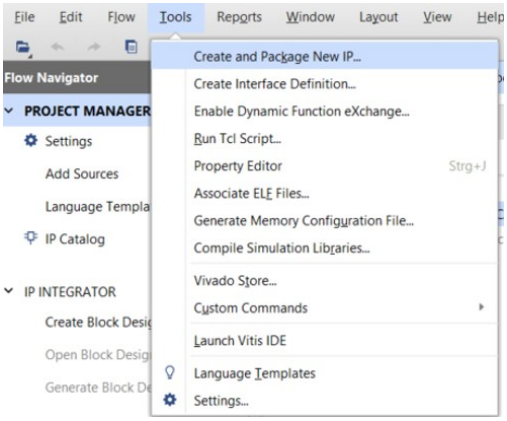
1.1.5 The next step is to create a block design.
→ Create Block Design → name the project



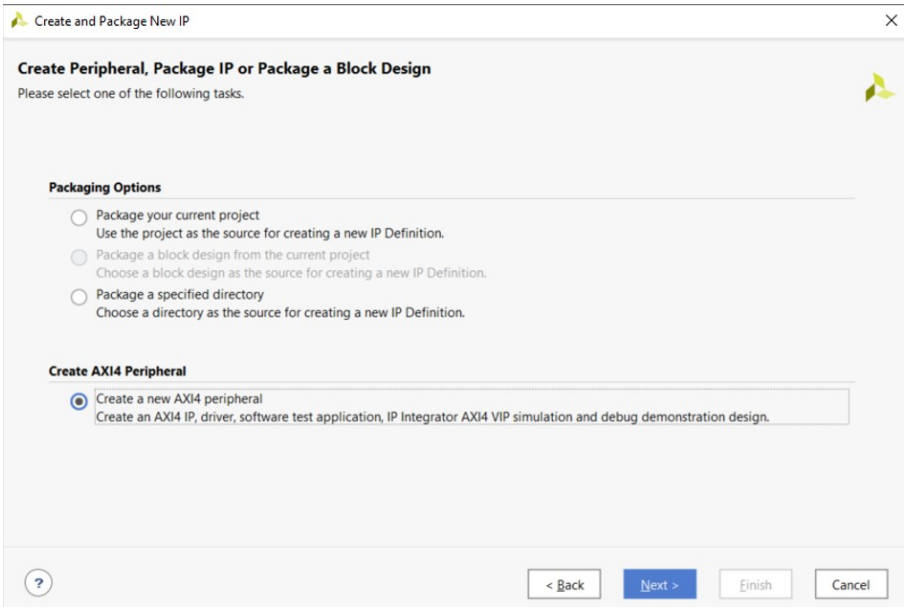
1.1.6 Theoretically, the design can now be created. The subsequent section will describe the process of creating one's own IP. If this is not required, the you can proceed to section 1.3.

1.2 Creation of an IP in Vivado

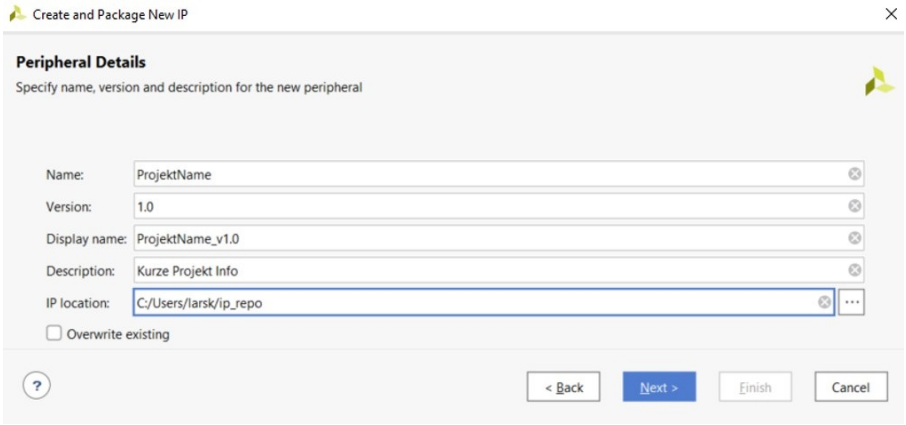
1.2.1 A new IP can be created if necessary. In this example, we are dealing with an Axi-Lite interface.
→ Tools → Create and Package New IP



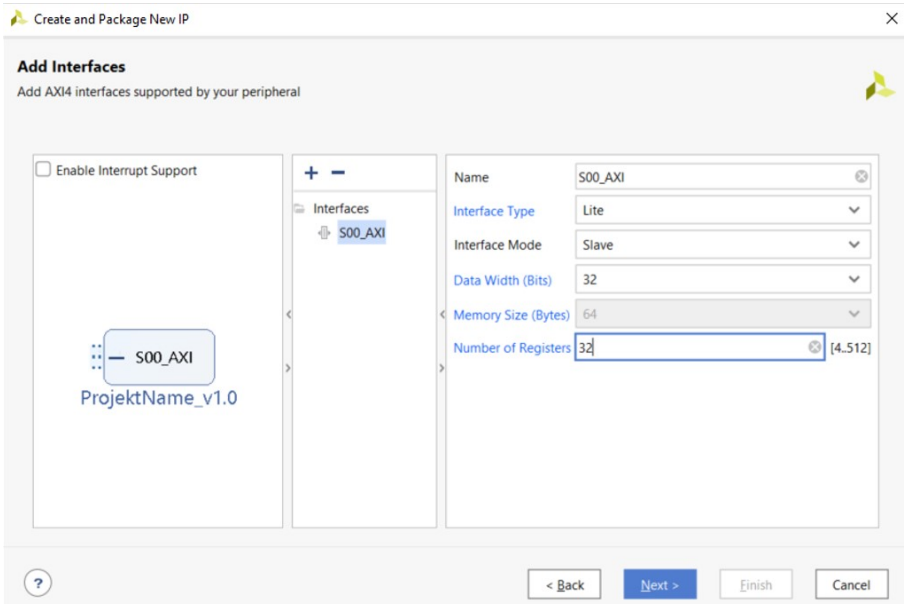
1.2.2 Choose Create a new AXI4 peripheral.



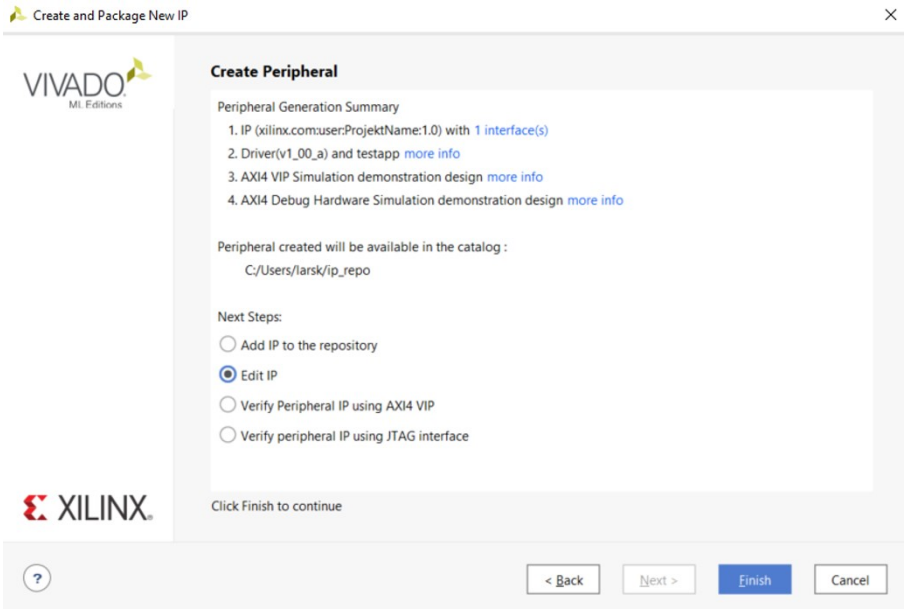
1.2.3 Assign a meaningful name and, if necessary, a short one-liner as information about the project.



1.2.4 The settings must be adjusted according to the project requirements.
→ Interface Type, Data Width, Number of Registers

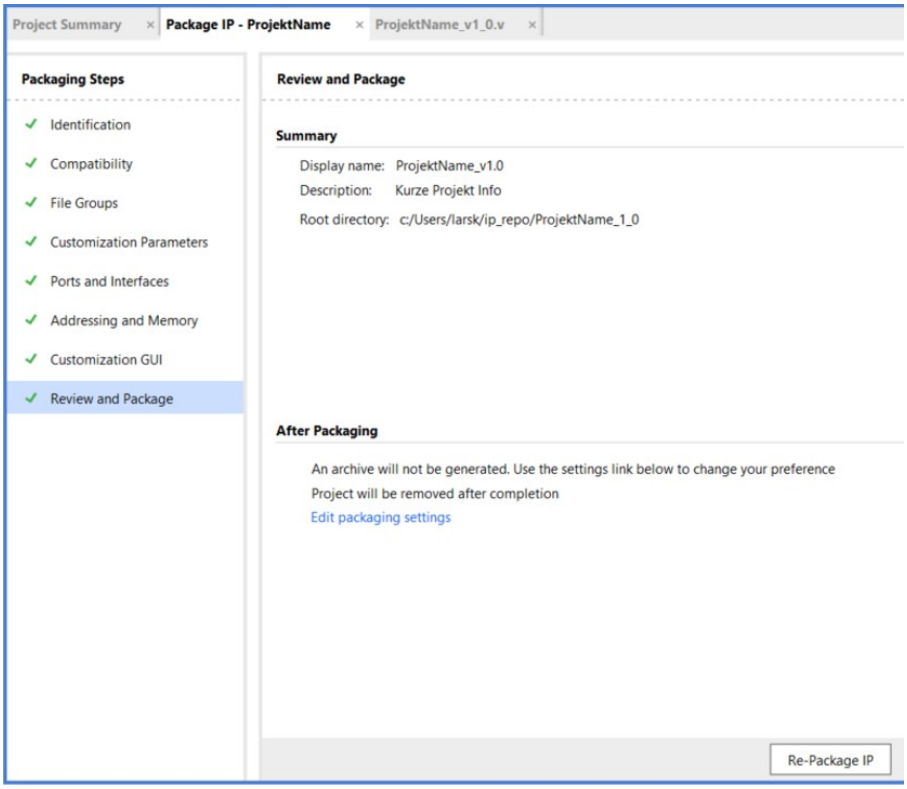


1.2.5 Edit IP directly in the next step.
→ Next Steps: Edit IP → Finish



1.2.6 The project can now be customized in the opening area. You can insert or add your own program code as well as inputs and outputs. The interface communication can remain the same or be replaced by your own. However, it should be easiest to adapt the register code in the second part of the program. However, this must be done according to the requirements and the project.

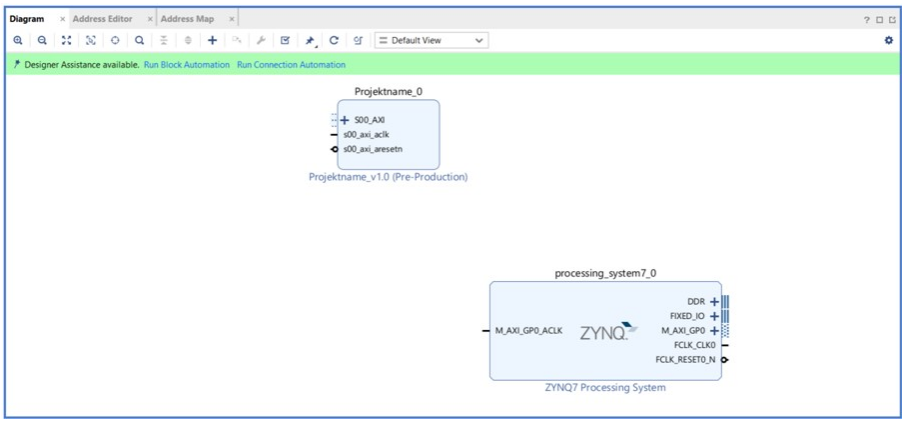
1.2.7 When the code has been modified as required, check in the Package IP window if all the check marks are green or if there are still changes to be made. This is confirmed with Re-Package IP. The IP window is automatically closed and the project is saved.



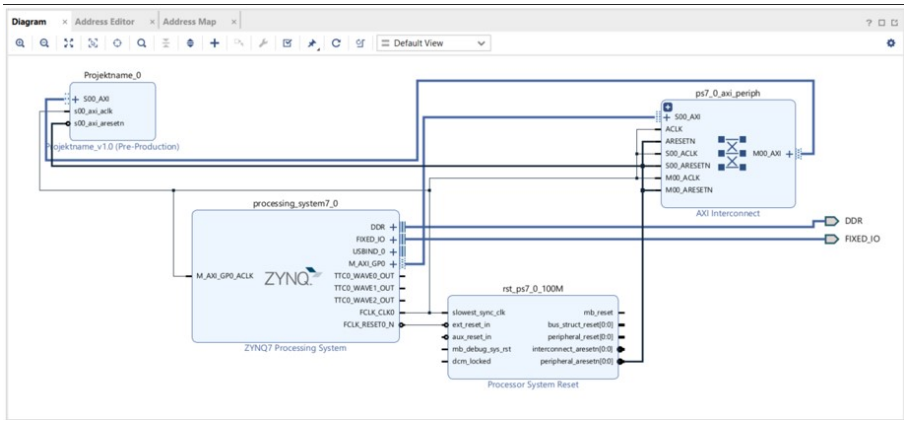
1.2.8 The IP created is then directly available. If this is not the case, the path where the project is saved should be checked or saved as IP access.

1.3 Create Block Design

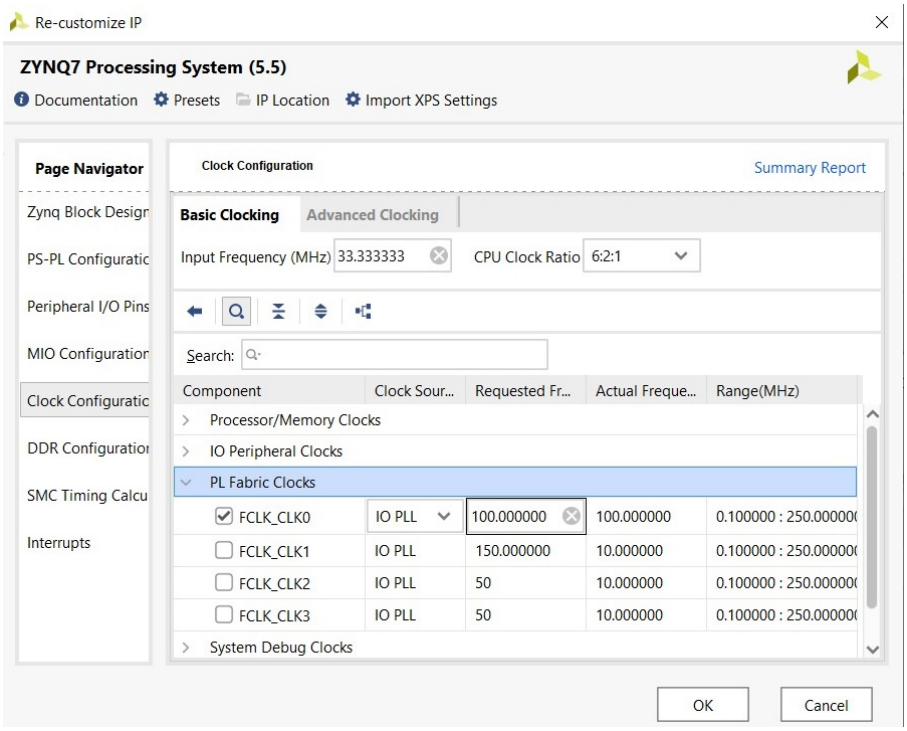
1.3.1 The block design can now be created. To do this, add the modules using the + symbol.



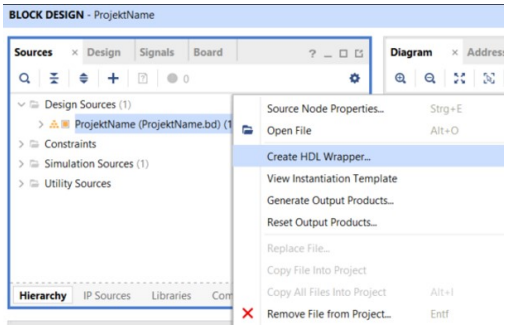
1.3.2 Once all modules have been added, Vivado can perform the connection automatically.
→ Run Block Automation
→ Run Connection Automation



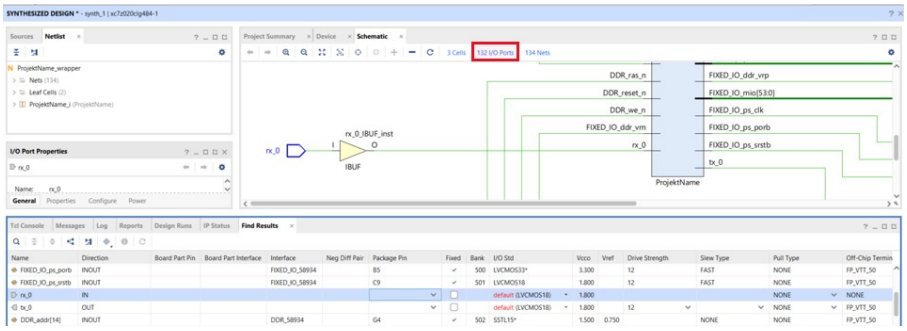
1.3.3 Then check the settings of the module you are using. This is a Zynq module. The clock should be set to 100MHz.
→ Right click on the Zynq Block → Customize Block



1.3.4 If all settings are as required, an HDL wrapper can be created
→ Right click on the project → Create HDL-Wrapper



1.3.5 Next, the synthesis must be performed.
→ Run Synthesis
1.3.6 If there are no errors, open the design and go to the schematic.
→ Open Synthesized Design → Open schematic with click on I/O Ports



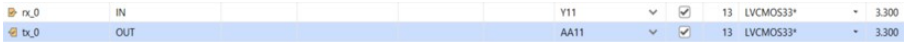
1.3.7 If input and output ports are required, they are now assigned to pins on the board. The assignment can be found in the board documentation..
→ Pins JA1 and JA2 on the Zedboard are used here as an example..

Table 16 - Pmod Connections

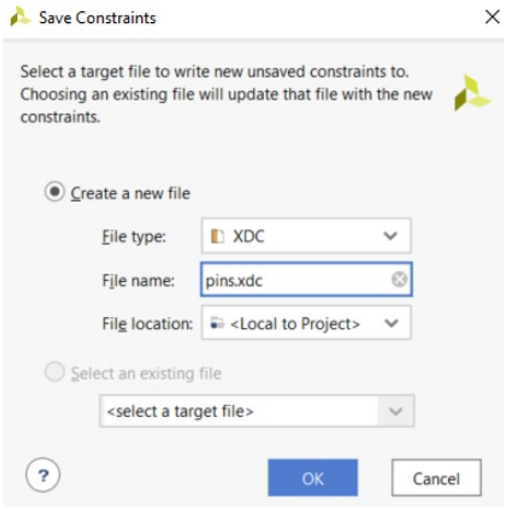
Pmod	Signal Name	Zynq pin	Pmod	Signal Name	Zynq pin
JA1	JA1	Y11	JB1	JB1	W12
	JA2	AA11		JB2	W11
	JA3	Y10		JB3	V10
	JA4	AA9		JB4	W8
	JA7	AB11		JB7	V12
	JA8	AB10		JB8	W10
	JA9	AB9		JB9	V9
	JA10	AA8		JB10	V8

Pmod	Signal Name	Zynq pin	Pmod	Signal Name	Zynq pin
JC1 Differential	JC1_N	AB6	JD1 Differential	JD1_N	W7
	JC1_P	AB7		JD1_P	V7
	JC2_N	AA4		JD2_N	V4
	JC2_P	Y4		JD2_P	V5
	JC3_N	T6		JD3_N	W5
	JC3_P	R6		JD3_P	W6
	JC4_N	U4		JD4_N	U5
	JC4_P	T4		JD4_P	U6

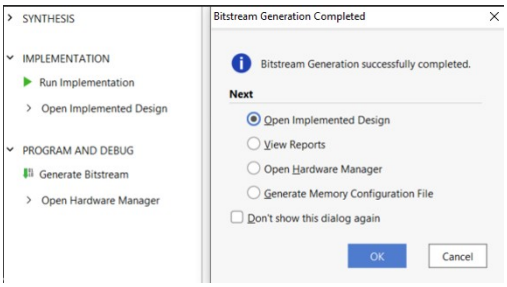
→ rx and tx must therefore be assigned to pins Y11 and AA11 on the board. Due to the higher signal voltages the field `I/O Std` LVCMOS33* has to be set correctly.



1.3.8 The new settings must be saved before the design is closed.
→Assign a meaningful name, e.g. pins.xdc

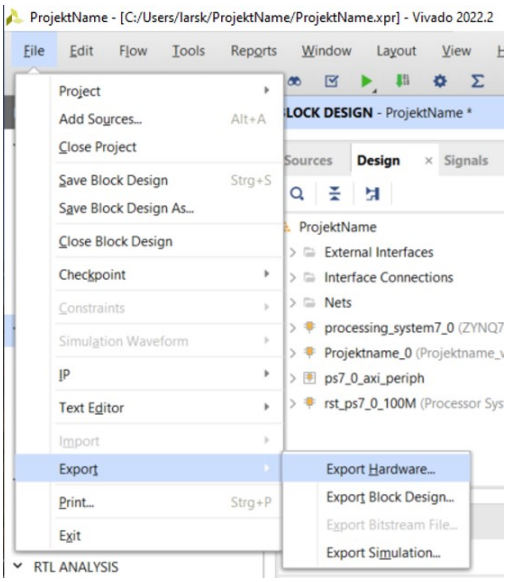


1.3.9 The bitstream can then be generated.
→ Generate Bitstream

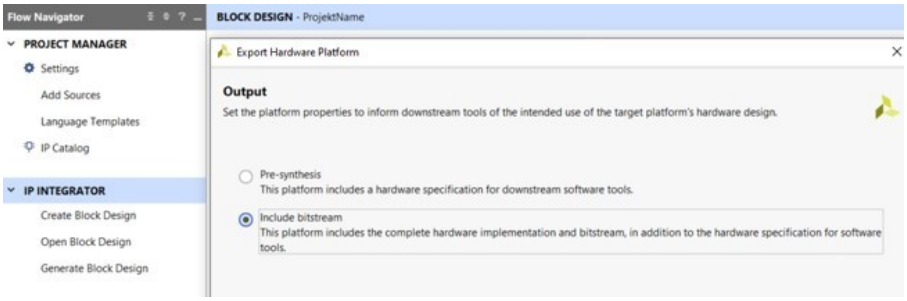


1.4 Export Hardware

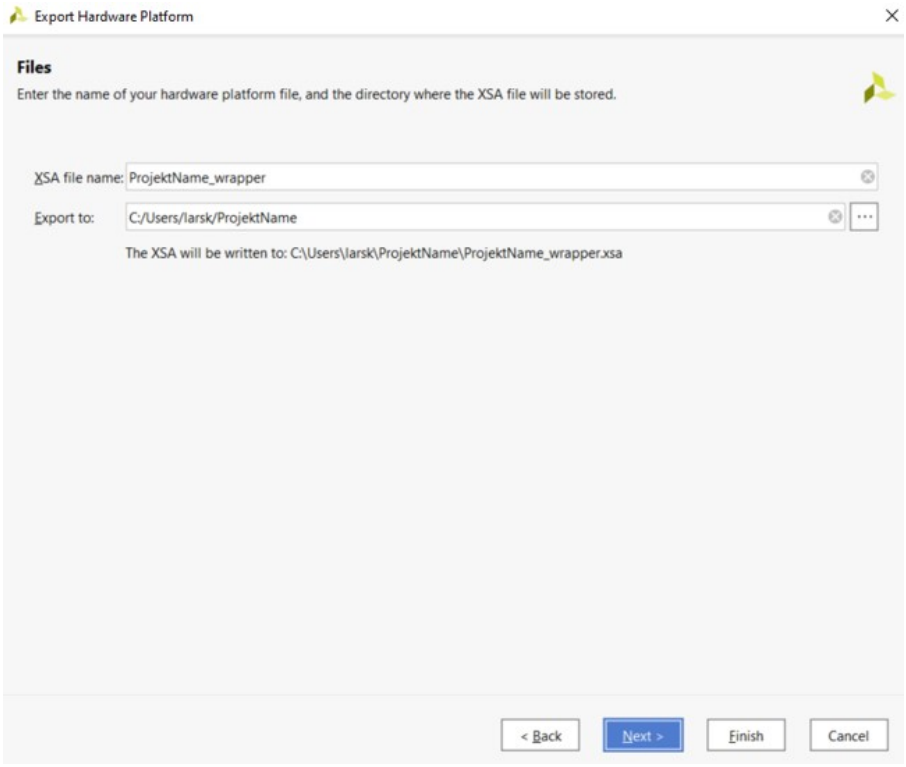
1.4.1 Once the bitstream is generated without errors, the design can be exported.
→ File → Export Hardware



1.4.2 Please make sure that the bitstream is included (Include bitstream).



1.4.3 You can now assign a name for the file to be saved in the project folder.



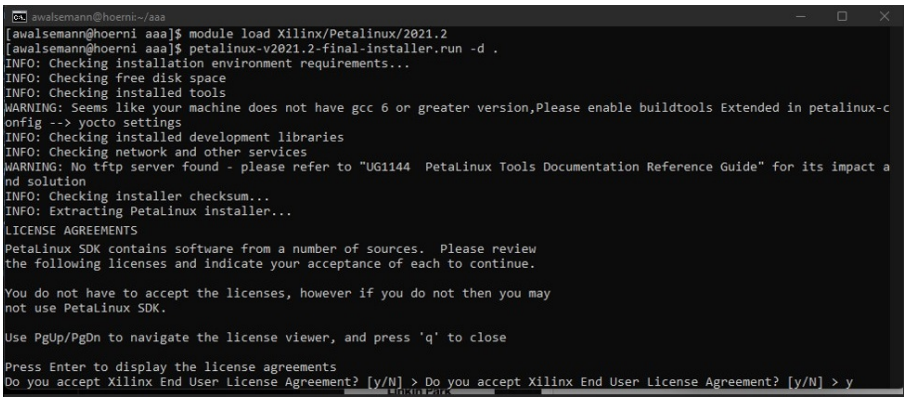
→ The project is now available as an .xsa file and can be used for the PetaLinux installation

2 PetaLinux-Installation

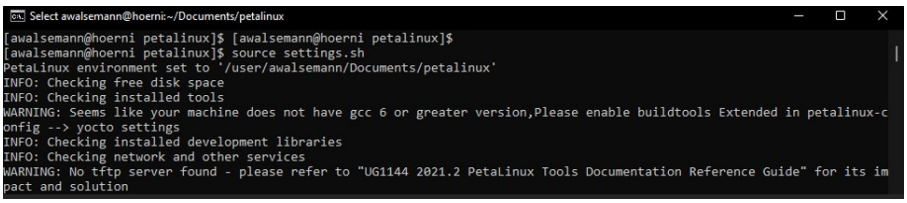
The installation process takes place under Linux on the lab computer. Commands are entered and executed in the terminal. Most steps can also be done via Remote Desktop. Only the creation of the data medium with the insertion of the root file can only be done directly on the PC where the data is located. Subsequent changes can then be made again via Remote Desktop if only the program code is changed and the sdk file does not need to be modified.

2.1 Installation of the PetaLinux environment (one-time implementation)

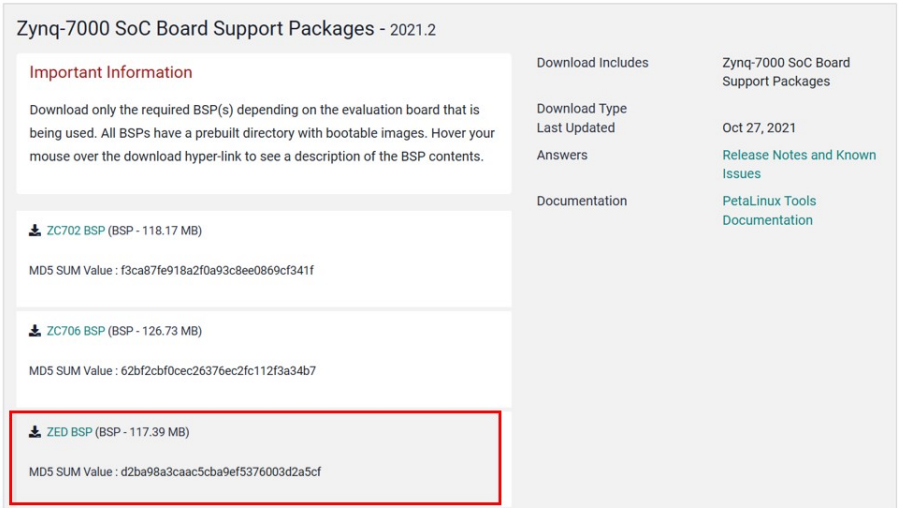
2.1.1 First open the input terminal and connect to the hoerni server. The installation process will then take place via this connection.
→ ssh hoerni
2.1.2 Login with the user password of your lab account.
→ Password
2.1.3 Now you need to load the appropriate environment module, install PetaLinux, and accept the license agreement.
→ module load Xilinx/Petalinux/2021.2
→ petalinux-v2021.2-final-installer.run -d



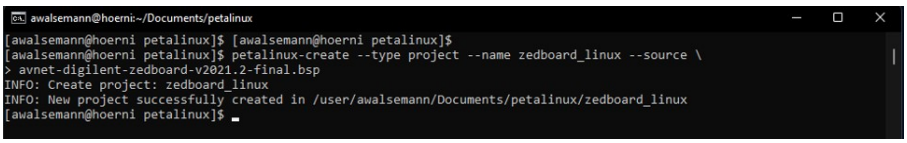
2.1.4 The location where the installation files will be stored is then created. This can be in the path of your home directory, for example. The name here is simply chosen as `petalinux`.
→ mkdir <user_home_dir>/petalinux
(<user_home_dir> names the path to the users home directory)
2.1.5 The next command makes the necessary PetaLinux commands available
→ source settings.sh



2.1.6 A file must now be downloaded from the Xilinx website and installed. This is a board support package for the FPGA board used (Zedboard).
Direct link to file: Board support Package
`https://www.xilinx.com/member/forms/download/xef.html?filename=avnet-digilent-zedboard-v2021.2-final.bsp`
Link via website, choose the red marked file shown in the example.
`https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools/2021-2.html`



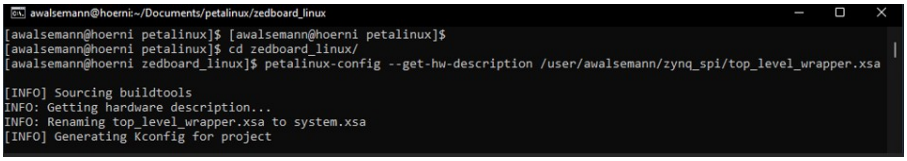
2.1.7 The file must be placed in the PetaLinux folder, and the subsequent installation must be performed in that folder. The command `mkdir` creates a new subfolder where the following installation files will be placed. A new project folder is created and the board file is loaded and installed with the following command. The name of the subfolder is `zedboard_linux`. The file contains e.g. information for the bootloader.
→ petalinux-create --type project --name --source avnet-digilent-zedboard-v2021.2-final.bsp



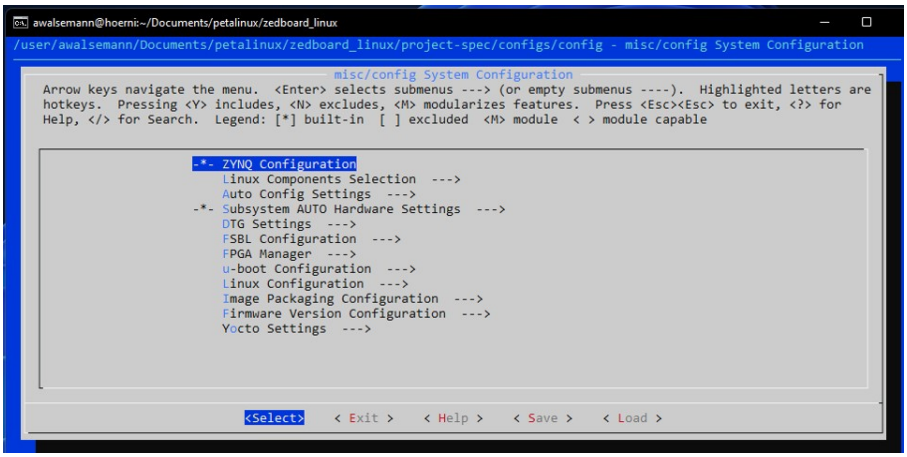
2.1.8 The license agreements must be confirmed again in order to continue.

2.2 Configure the project

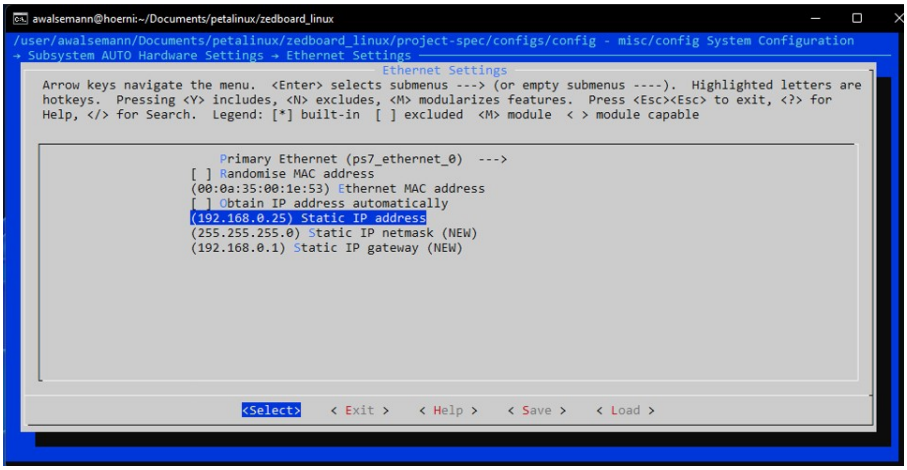
The following steps can now be repeated whenever the project has been changed or a new one is created. If necessary, the connection to the server must first be established, with `ssh hoerni` and login using your own password. Please take care about the folder you are using! [3mm] 2.2.1 The hardware design is now imported and previously saved in the project folder. The next steps can be done in this folder. In this example this is the `zedboard_linux` folder.
→ cd zedboard_linux
2.2.2 Now the PetaLinux installation will be done with the hardware design and the .xsa file will be included.
→ petalinux-config –get-hw-description Projektname_wrapper.xsa



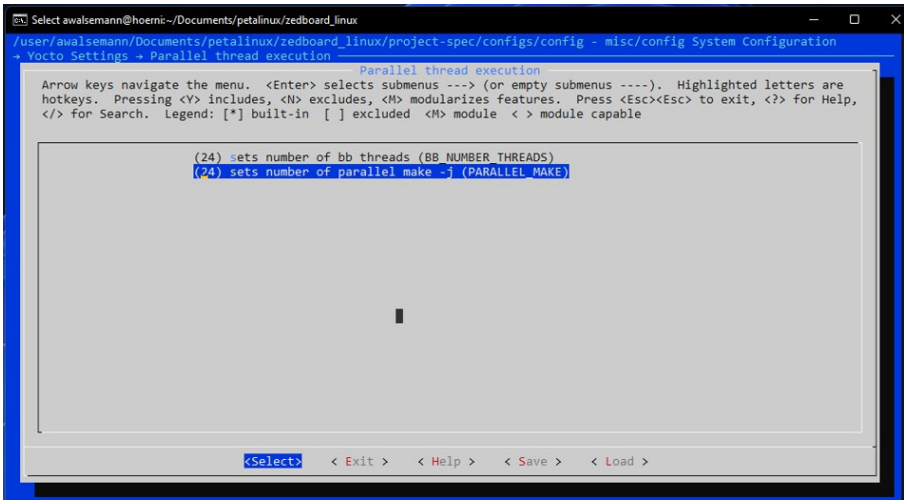
2.2.3 The installation takes a moment and then a configuration window opens. Here you can configure the system. The following settings should be made, but can be tweaked as needed.



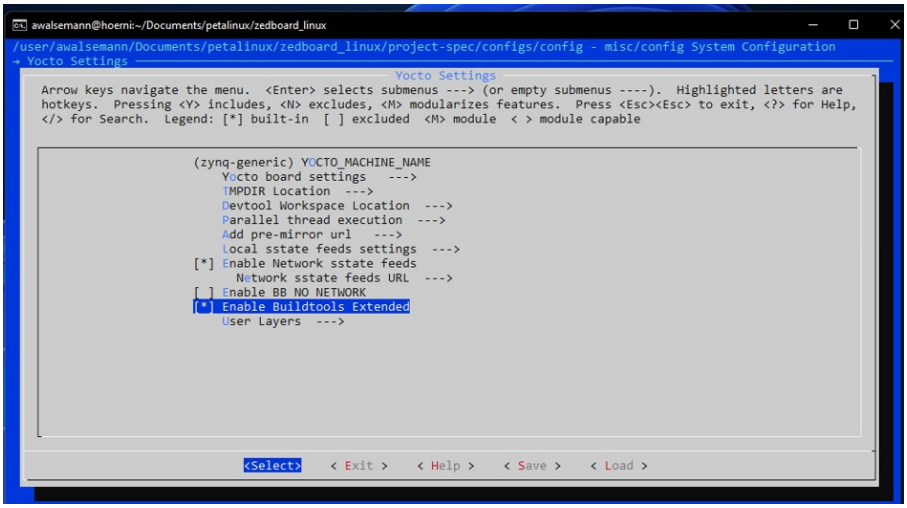
2.2.4 Specifying the IP address
→ Subsystem AUTO Hardware Settings → Ethernet Settings → Static IP address



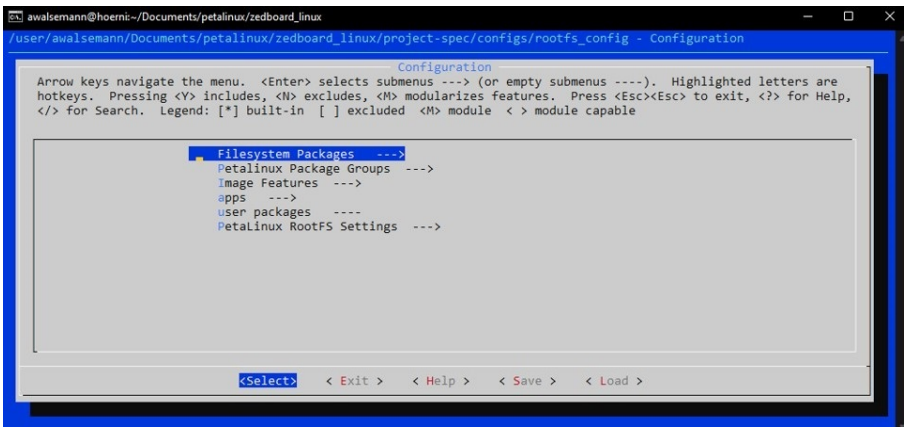
2.2.5 The parallel thread execution is set to 24.
→ Yocto Settings → Parallel thread execution



2.2.6 Enable Buildtools Extended is set. (This is a better adapted version of Xilinx for this application)
→ Yocto settings → Enable Buildtools Extended → Y zum Auswählen

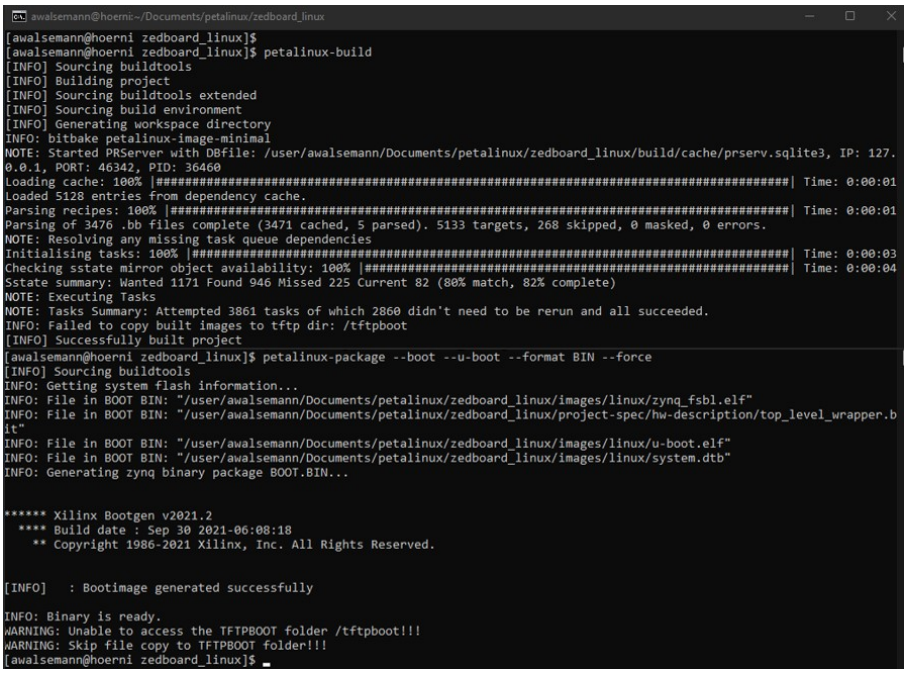


2.2.7 Exit settings with EXIT and save with Y.
2.2.8 The rootfs config is now editable. This can be accessed in the terminal with the following entry.
→ petalinux-config -c rootfs
2.2.9 Add Peek and Poke to read and write registers.
→ apps → peekpoke → choose with Y



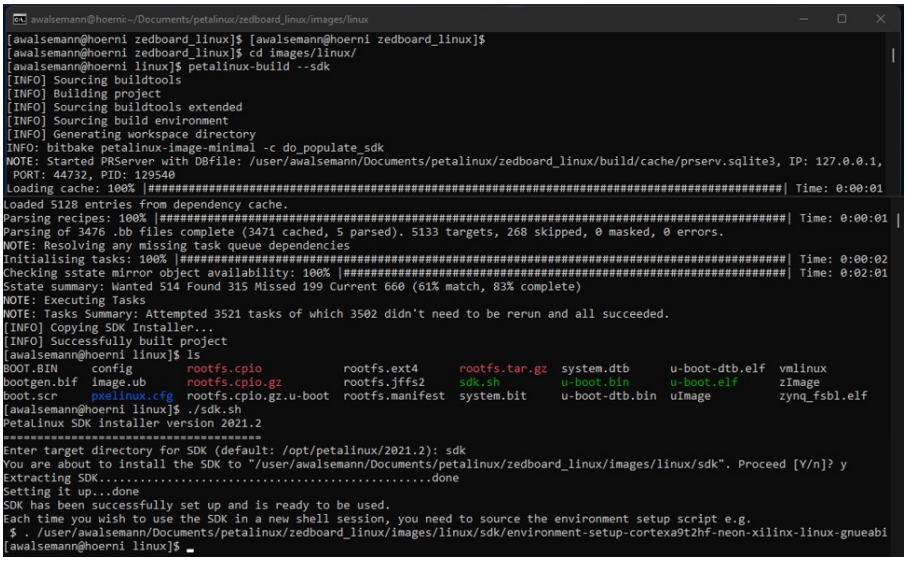
With peek and poke, data can be read from or written to the memory at a desired address for test purposes.

2.2.10 The next command in the terminal creates the project and saves the program files.
→ petalinux-build
→ petalinux-package --boot --u-boot --format BIN --force



The created files are located in the folder zeboard_linux/images.

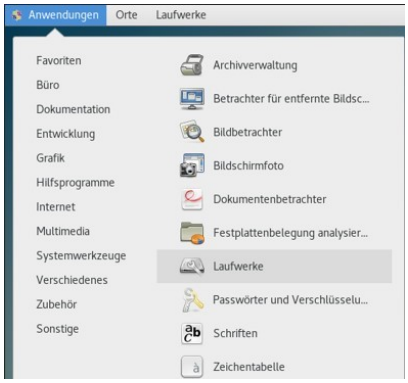
2.2.11 The last step is to generate the SDK for later use in Vitis in the Linux folder.
→ cd images/linux/
→ petalinux-build –sdk



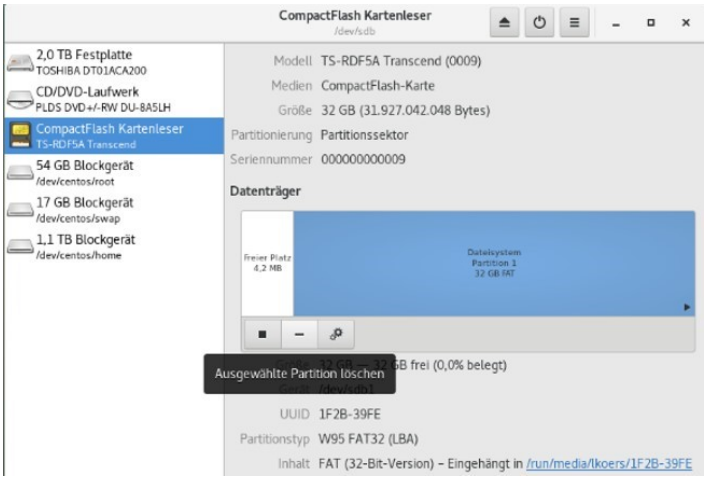
2.2.12 The sdk file only needs to be reloaded later if necessary, e.g. for new drivers. If only the program code has changed independently of the further requirements for Vitis, it is sufficient to reload the `BOOT.BIN`, `BOOT.src` and `u_boot.ub` files onto the SD card.

3 Prepare SD card for boot files

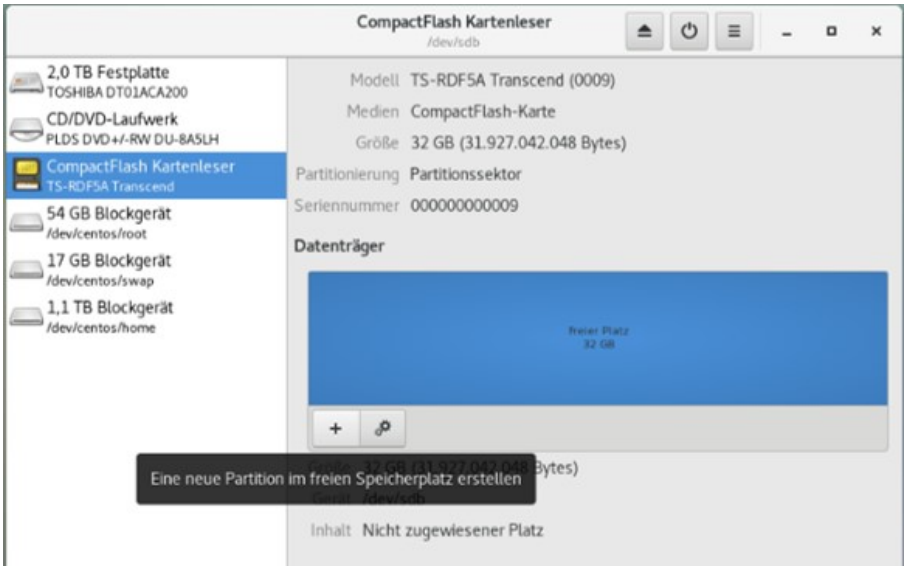
3.1.1 The SD card must be split into two partitions. The division is relatively arbitrary, but the `boot` partition should always be larger than the `root` partition.
→ Open drivers (Laufwerke)



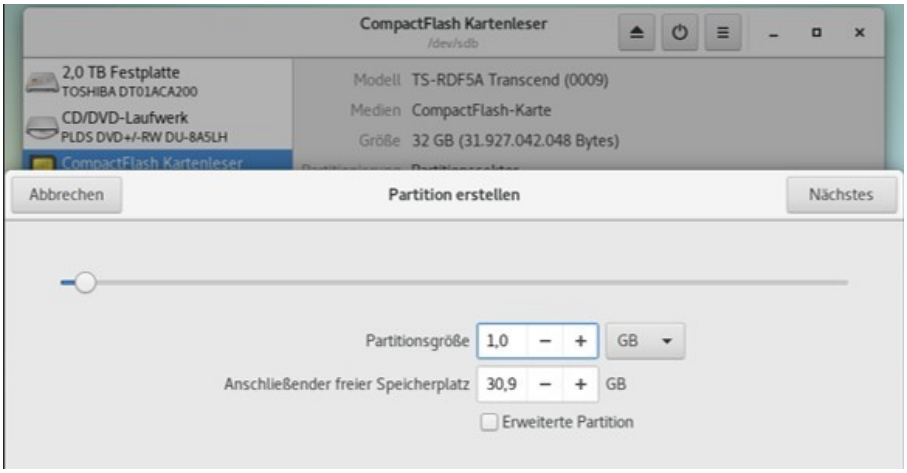
3.1.2 Divide into two partitions, if necessary delete an existing one with -.



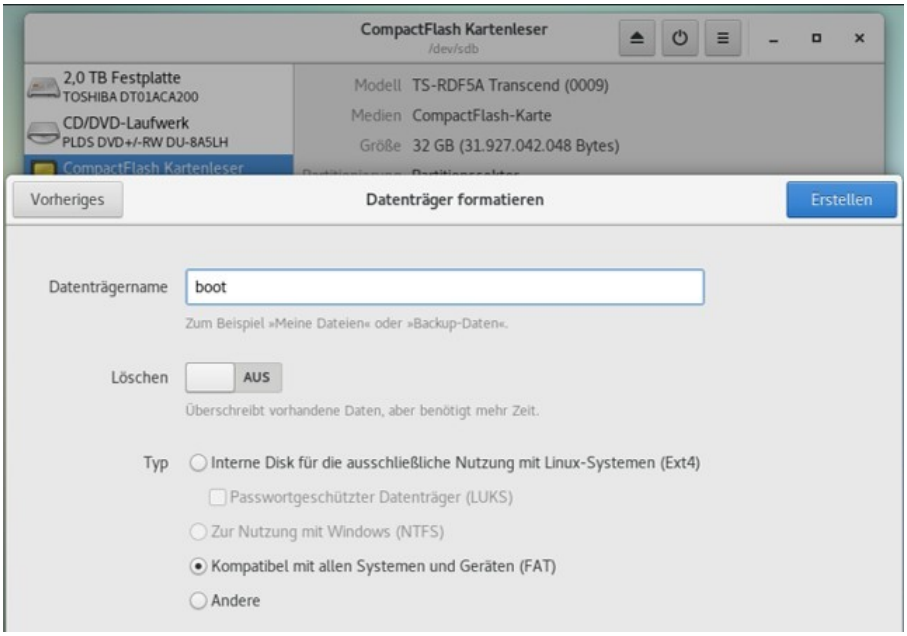
3.1.3 Next, add a new partition with +.



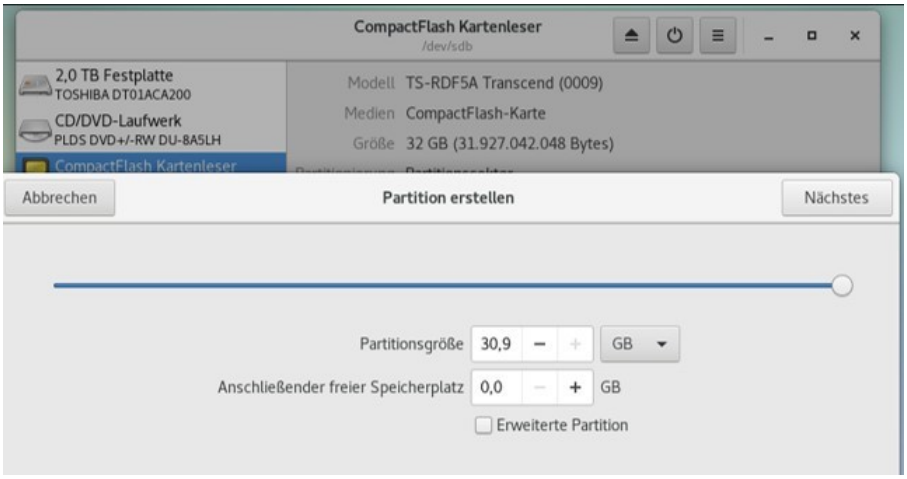
3.1.4 Define memory allocation. e.g. 512MB or 1GB for the first partition.



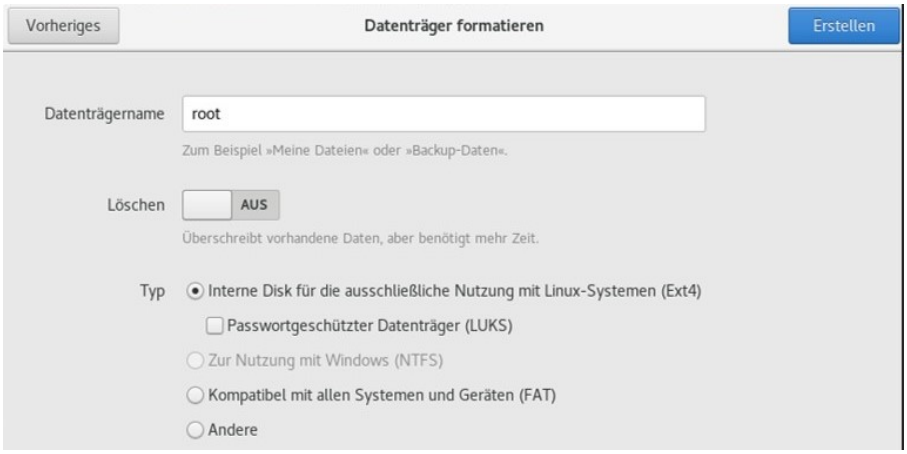
3.1.5 Specify the name of the partition with boot and leave type FAT selected.



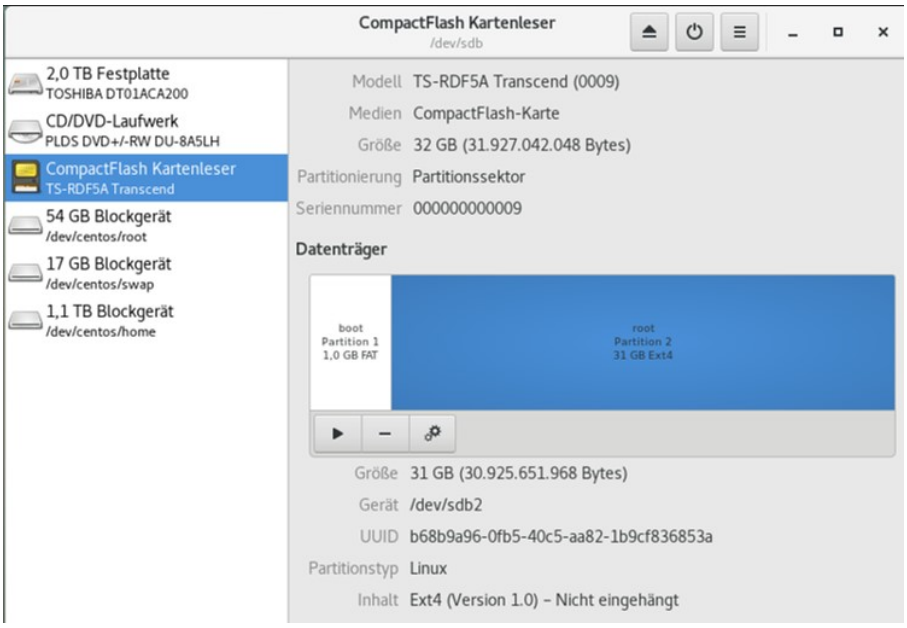
3.1.6 Create next partition with + and assign available remaining memory to the partition.



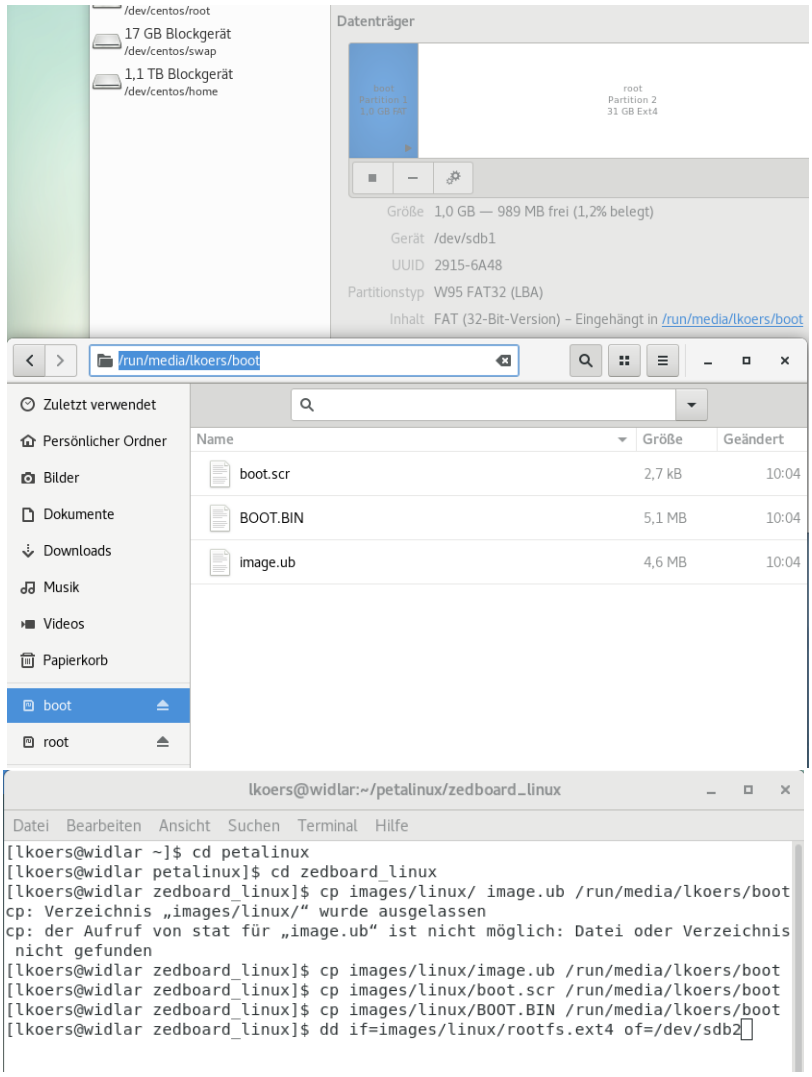
3.1.7 Assign the name root and select type Ext4.



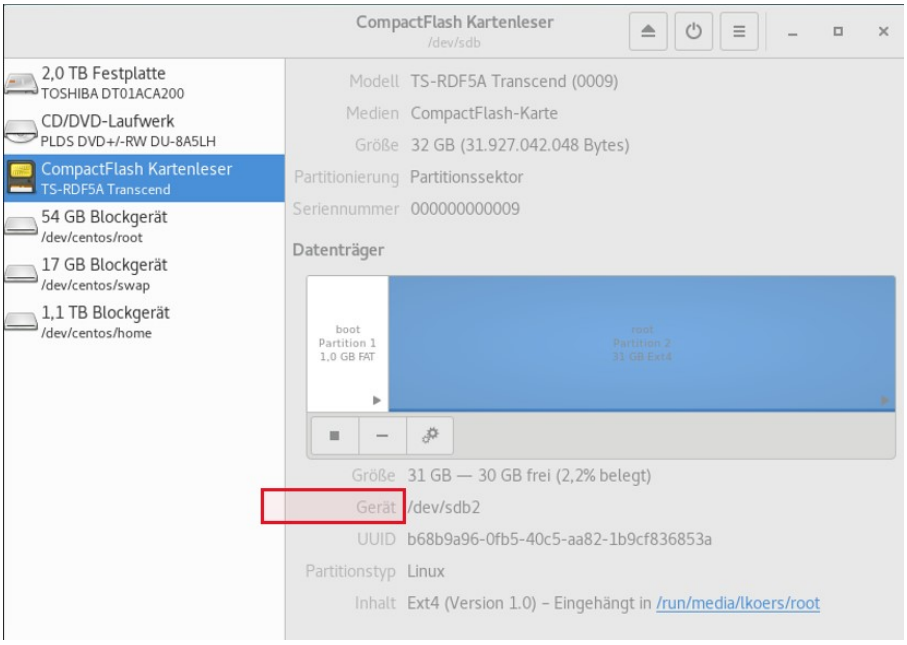
→ Completed partitions on the SD card.



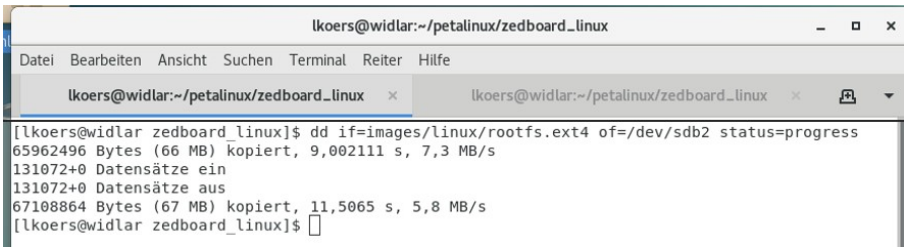
3.1.8 The previously created files can then be loaded onto the SD card. It is sufficient to copy three files to the boot partition.
→ Either copy and paste
Or by a command in Linux with the filename and path of the memory card (blue).
→ cp images/linux/ filename /run/media/user/boot



3.1.9 The rootfs file **must** be moved as described in the next step!
Run the command in a Linux terminal!
!!!Attention move to correct path, root folder → See SD card next to device (marked in red)



→ dd if=images/linux/rootfs.ext4 of=/dev/sdb2



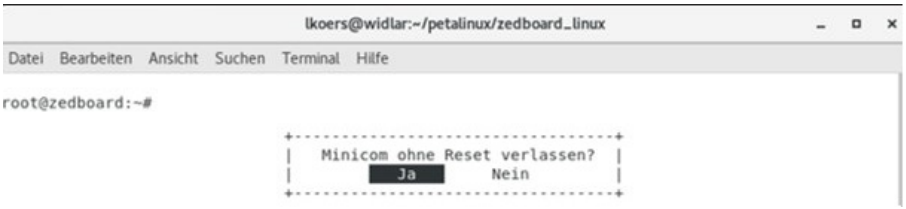
3.1.10 After executing the command, the data will be on the SD card. Please check whether there really is anything in the respective partitions. If necessary, remove and reinsert the SD card once.

4 Test the finished installation on the SoC board

4.1 Test under Linux

4.1.1 Once all the steps have been completed, the SD card can be inserted into the Zedboard, the board connected to the PC and switched on. If the installation process is correct, the project will boot and can be used.
→ Turn on → Boot process starts automatically
4.1.2 Check the connection under Linux. To do this, open the terminal window and enter the following command:
→ minicom -D /dev/ttyACM0

4.1.3 Among other things, peek and poke commands can be executed here for testing purposes.
4.1.4 If the program is to be terminated, this can be done as follows.
→ Strg a → then q to leave the application



4.2 Test under Windows with Tera Term

Commands can be sent to the board via a program such as `Tera Term` and the responses read.

4.2.1 To do this, open the program and select the COM port to which the board is connected. The boot process can then be viewed as in the example here and commands such as peek and poke can be used at the end.

5 Reinstallation in case of e.g. adjustments in the source code

5.1.1 If errors have crept in, you can always reinstall on the server. The commands needed to do this are briefly listed here. These are sufficient to make a change that does not affect the image or the sdk file. After the Vivado project has been exported again as an .xsa file, the file can be placed in the PetaLinux folder.
This process can now be done via remote desktop, for example.

5.1.2 Now you need to enter the following commands in the Linux terminal:

!Please note that the texts in italics are project-specific!

→ssh hoerni
Login with password
→cd *petalinux*
→source settings.sh
→cd *folder-of-.xsa-file*
→petalinux-config --get-hw-description *Projektname_ wrapper.xsa*
Exit when configuration settings are opened
→petalinux-build
→petalinux-package --boot --u-boot --format BIN --force

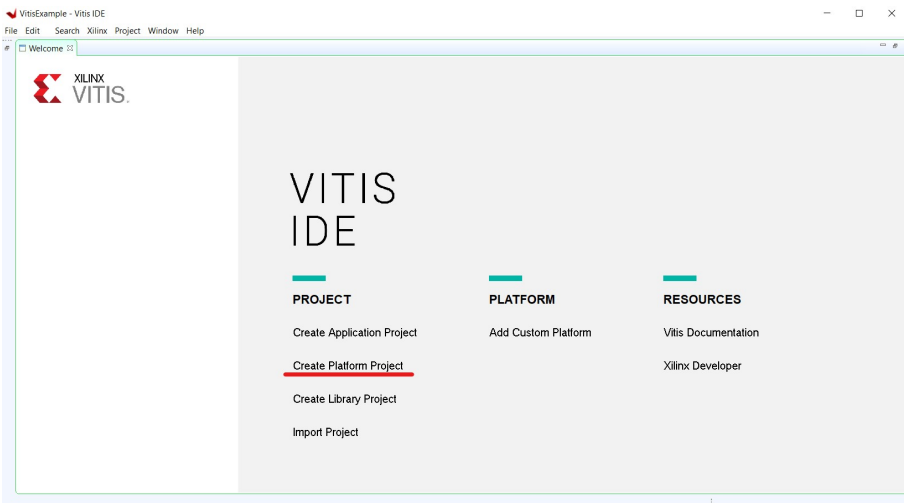
5.1.3 Then copy the three files (`BOOT.BIN`, `BOOT.src` and `u_boot.ub`) for the boot folder again and drag them to the SD card.

6 Integration of the hardware design in a Vitis platform project

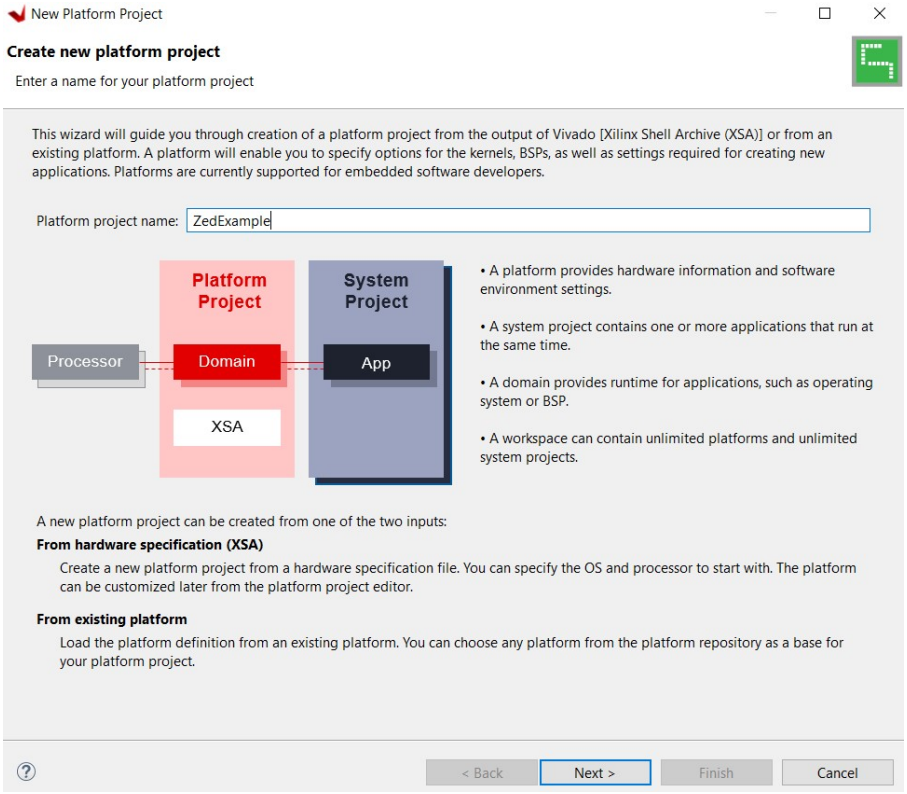
The following steps are done in Vitis 2022.2.

6.1 New Hardwareplatform

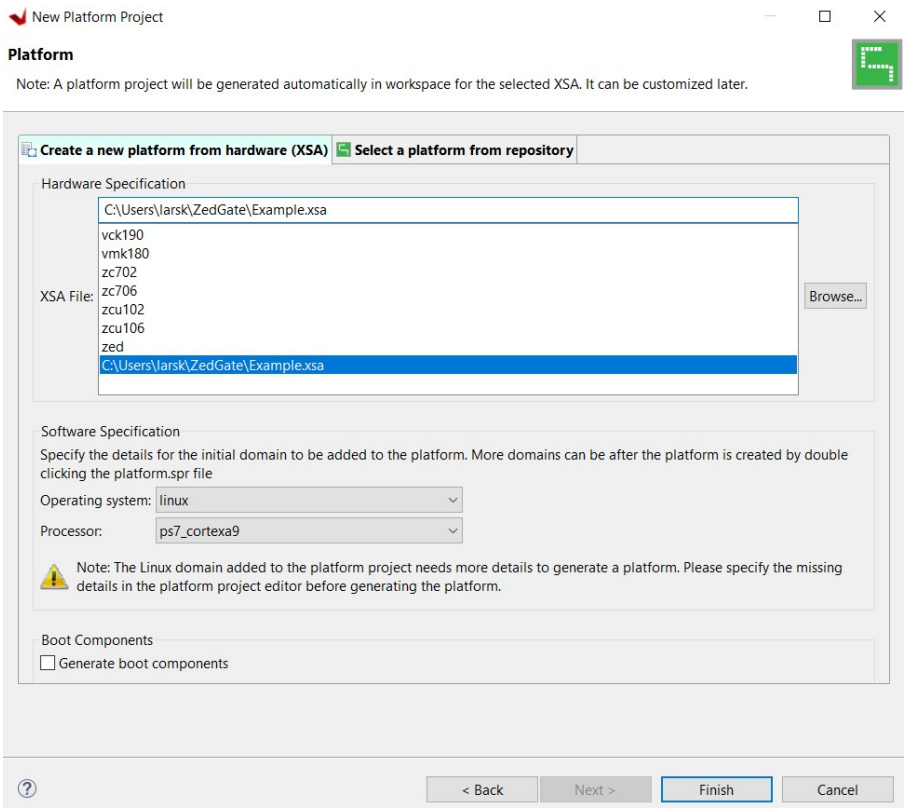
6.1.1 To start, a new platform project has to be initialised.



6.1.2 Choose a name for the platform project and go on.

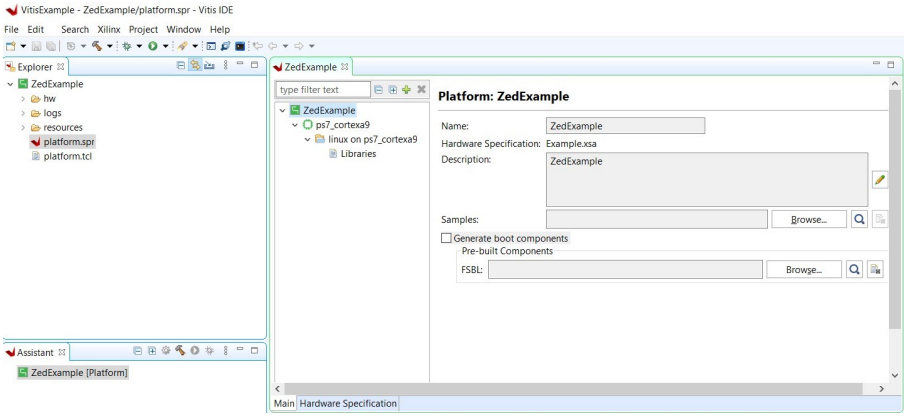


6.1.3 The .xsa file, that is created from the Vivado-project, has to be selected.
→ set operating system to linux
→ take care, that the field *Generate boot components* is **not** set!

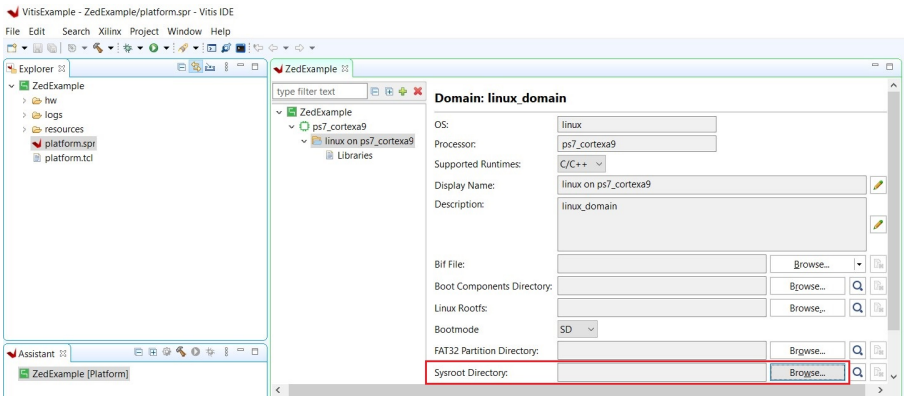


→ Finish the new platform project.

6.1.4 After finishing, the platform project will open up.



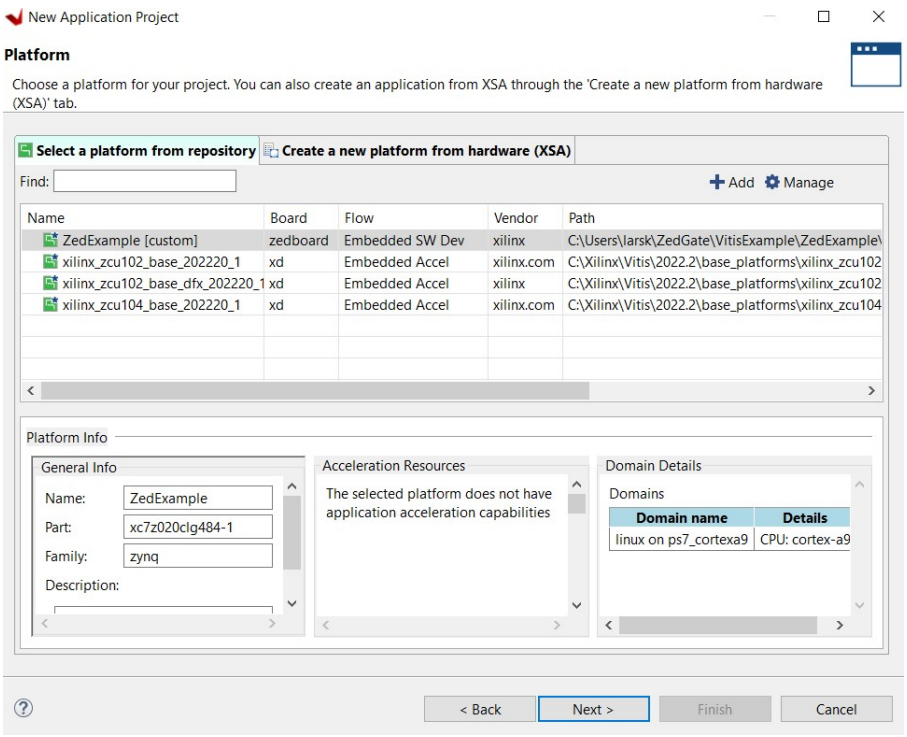
6.1.5 Go to the folder: *linux on ps7_cortexa9*
→ set the Sysroot Directory to the images path from the previous Linux installation, see for example:
→ petalinux/zedboard_linux/images/linux/sdk/sysroots/cortexa9t2hf-non-cilins-linux-genueabi



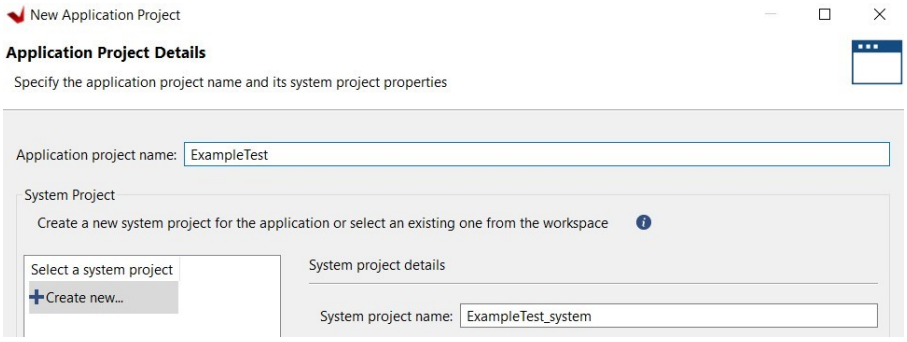
6.1.6 Build the platform project! If no errors occur, the platform is ready to be used for an application project.

6.2 Create Application Project

6.2.1 Create a new application project and select the generated custom platform.

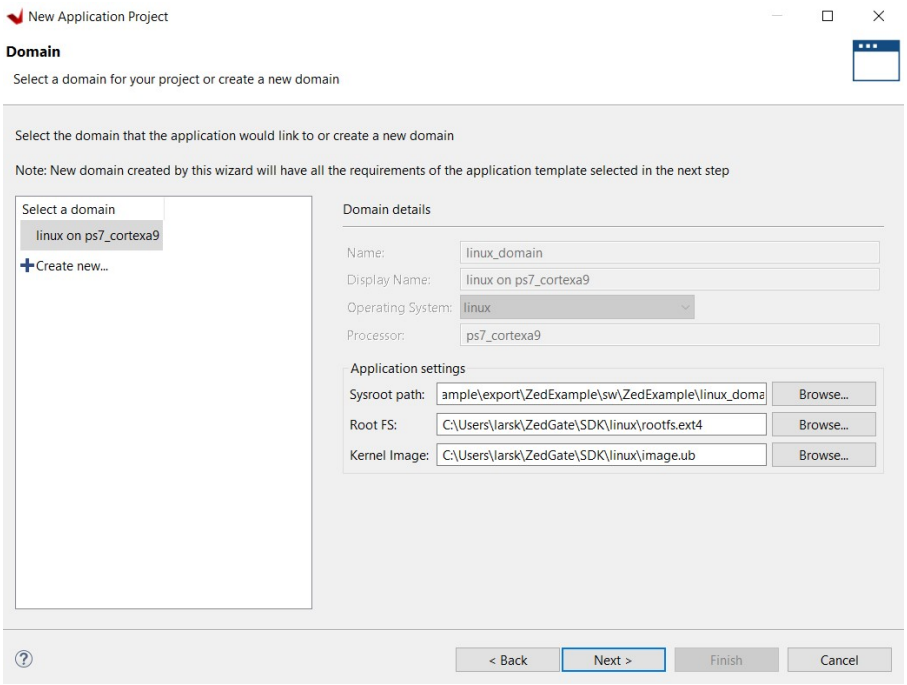


6.2.2 Give a name to the project and go on.



6.2.3 Set the file-path of the following files:
The Sysrooth path points to the workspace of your Vitis project.
→ Sysrooth path:
VitisExample/ZedExample/export/ZedExample/sw/ZedExample/linux_domain/sysroot/cortexa9t2hf-neon-xilinx-linux-gnueabi

The other two files are located in the installation path of your PetaLinux installation.
→ Root FS: ...LINUXPATH.../linux/rootsfs.ext4
→ Kernel image: ...LINUXPATH.../linux/image.ub

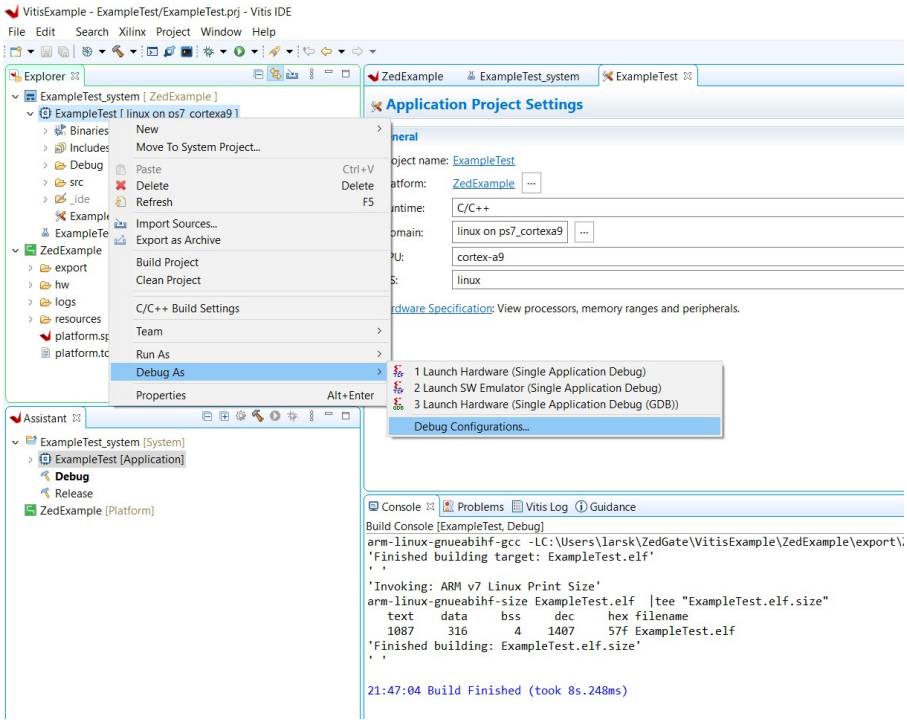


6.2.4 Go on and choose either an empty application or for example the “Hello World Application“.
→ Now Build the project and see if no errors occur.

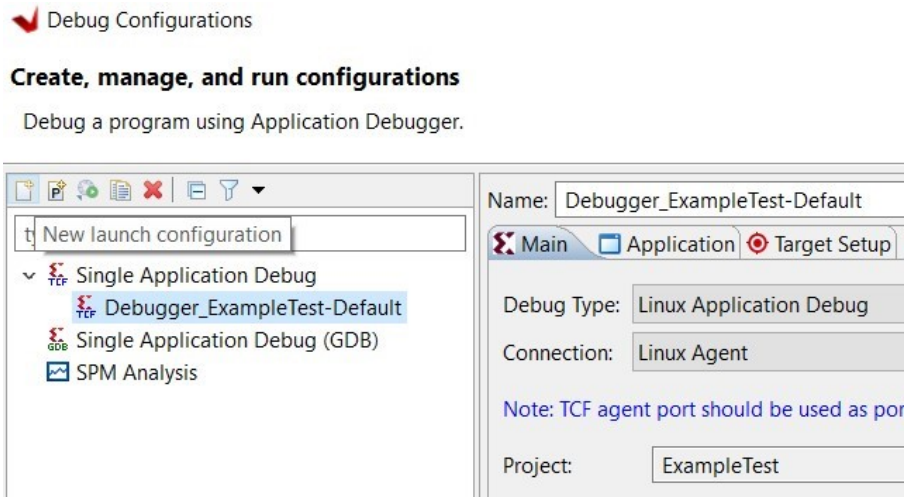
Please take care that the correct path of the files are selected. The shown pictures in this guide are just examplary.

Afterwards the system can then be started via launch hardware and use of an USB-cable. Carry out the following steps to set a destination with an IP address and a LAN connection:

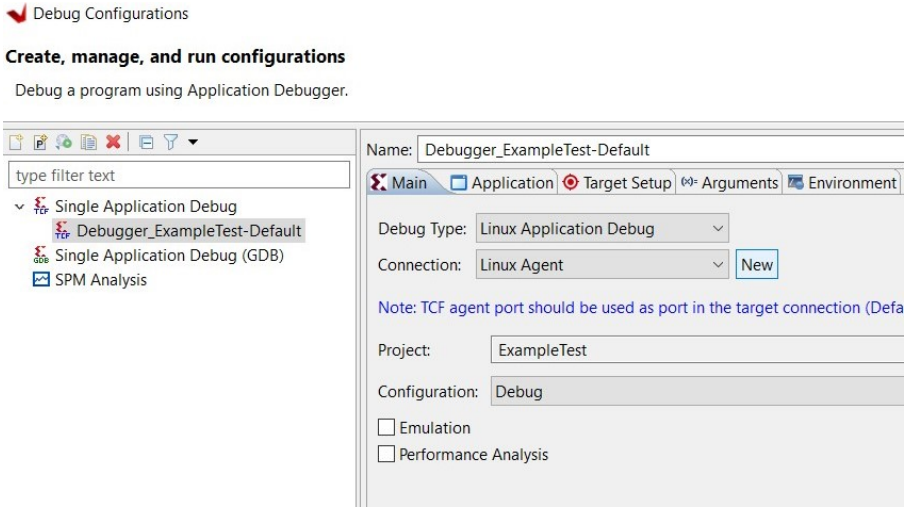
6.2.5 Open *Debug configuration* by clicking with the right mouse button on the application project. Go to *Debug as* and open the *Debug Configuration*.



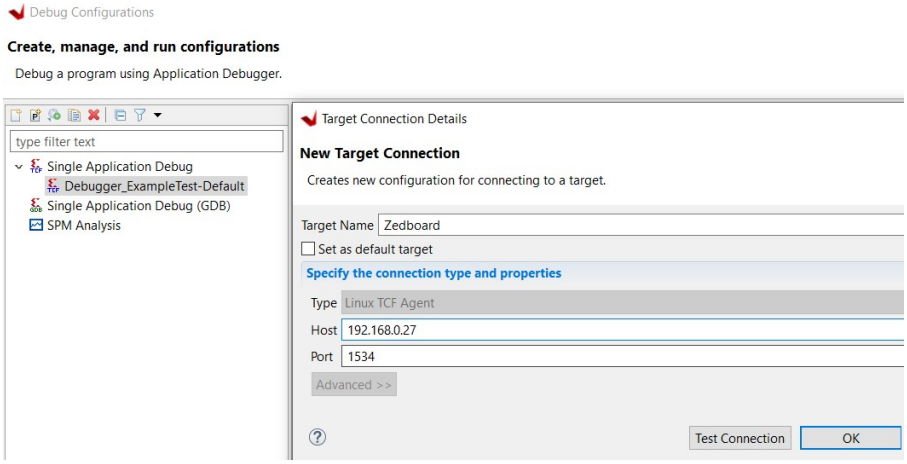
6.2.6 Add a new debug configuration by first clicking on *Single Application Debug* and then clicking on the "New Configuration" button, which is the paper with the plus symbol. This creates a new configuration with the name *Debugger*_PROJECTNAME-*Default*.



6.2.7 Click on *New* behind *Connection: Linux Agent*

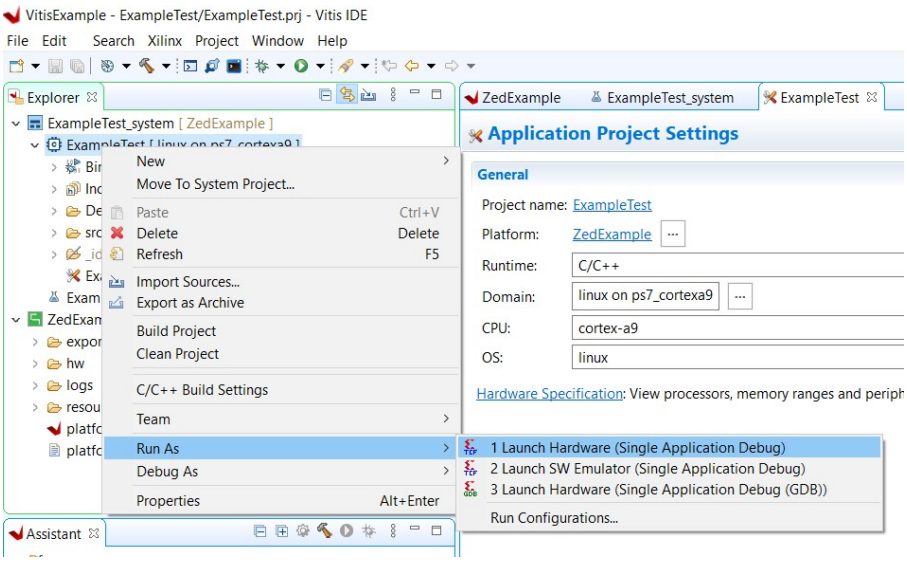


6.2.8 Type the IP address of the project that has been set while installing PetaLinux and close the settings.



Confirm the new settings with yes

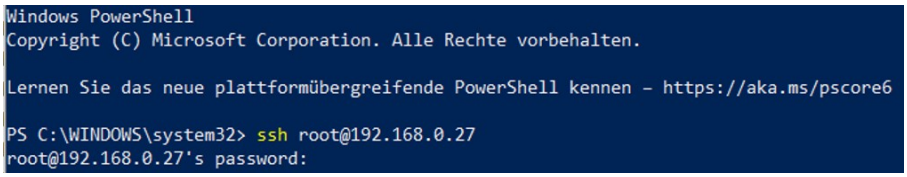
6.2.9 By using Run as → Launch Hardware, the project is now loaded on the connected device.



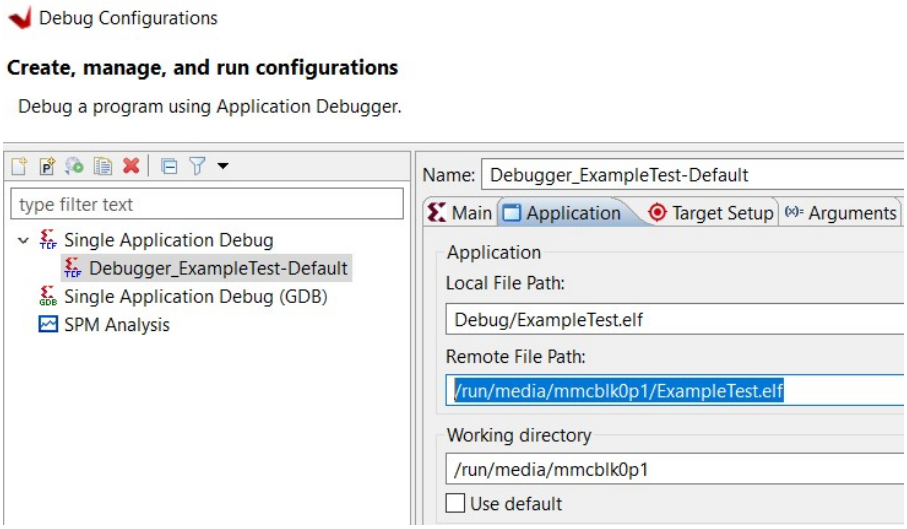
6.3 Use of Vitis Program in a Terminal

6.3.1 After the program has been loaded on the device, the system can be used. For this a terminal has to be opened and the connection has to be established.

6.3.2 Connect to the IP-Address of the application.
→ ssh root@192.168.0.IP
→ Use *admin* as password



6.3.3 Type in the *Remote File Path* from the application to start your program. This path can be found in the previous changed debug configurations under *Application*.



6.4 Use of Remote screen for sessions

6.4.1 The use of a remote screen is very useful, to disconnect from the device without cancelling the running program. To do so, the following commands have to be used after connecting to the device and before starting the program.
→ screen -R Example
Example is just a space holder and can be used as preferred. Just remember this name for later reconnection.
This command opens up a new screen, in which commands to start the system can be asserted.
6.4.2 To close the screen and disconnect without interrupting, these buttons have to be clicked together.
→ strg → a
And after that click
→ d
6.4.3 Now the screen is closed and the program still runs on the device. To open up the screen, the same procedure has to be done again with the name chosen during the initial start.

Acknowledgements

At this point I would like to thank some people who supported me during my master thesis.

First of all, I would like to thank Prof. Dr. Karagounis for giving me the opportunity to be a part of such an inspiring research institute as CERN for my master thesis. His knowledge and expertise contributed a lot to the thesis.

I would also like to thank Cologne Chip for providing the GateMate hardware for the tests, as well as the support in the realization of different design approaches during this time.

Many thanks to Rudy Ferraro and Salvatore Danzeca for making the internship at CERN possible and for the many insights. The support and guidance from them, Antonio Scialdone and the whole team made this time very instructive and unforgettable.

Finally, I would like to thank my family and friends for their support during my time away from home. Special thanks go to my wife Paula Koers for her unconditional support during this time, both from afar and on site.

April 2024, Lars Koers